HIP Internal Report Series
HIP-2021-03

Deep Neural Network Classifiers in CMS Track Reconstruction and in the Search for the Charged Higgs Boson

Joona Havukainen

Helsinki Institute of Physics
University of Helsinki

Doctoral dissertation, to be presented for public discussion with the permission of the Faculty of Science of the University of Helsinki in the auditorium CK112 at Exactum, Pietari Kalmin katu 5, Helsinki, on the 20th of October 2021 at 13 o'clock

Helsinki 2021

ISBN 978-951-51-1299-6 (print)
ISBN 978-951-51-1300-9 (pdf)
ISSN 1455-0563
Printed by Unigrafia Oy Yliopistopaino
Published electronically at ethesis.helsinki.fi
Helsinki 2021

Aknowledgements

This book is the culmination of the work done between autumn of 2016 and end of year 2020, as a member of the Compact Muon Solenoid (CMS) experiment at CERN and the CMS Experiment group at the Helsinki Institute of Physics (HIP) in the University of Helsinki. The main portion of this research work was funded by Jenny and Antti Wihuri foundation and multiple conferences and schools were funded by Waldemar Von Frenckell foundation, Magnus Ehrnrooth foundation and the Doctoral School of Natural Sciences. This support not only allowed me to focus on my research as well as become a part of the international network of scientists in my field and is greatly appreciated.

The unique data that all the work in this thesis builds on is the result of generations of researchers who have been involved in designing, building and operating both the Large Hadron Collider at CERN and the CMS experiment. These are some of the most sophisticated machines ever built by humankind and an awe-inspiring tour de force of international collaboration. I am grateful to have experienced working with the talented and passionate people involved in these collaborations throughout my research work.

A collaboration of people has also been necessary to reach the stage of defending a doctoral thesis and many people deserve my heartfelt gratitude for all the help and support I have received throughout this process. The biggest impact on the thesis has been from my thesis supervisor Prof. Paula Eerola and my thesis advisor Doc. Sami Lehti. While I have been given the academic freedom to pursue the paths of research I have felt most interesting and important, they have shepherded me towards the right direction when needed. No matter how distraught I have been with my work or my results, talks with Paula have always managed to get me to take a step back and see the bigger picture how I am progressing along the right direction. I managed to enter an extremely vibrant and fast-paced field of machine learning at an opportune time thanks to Sami's excellent sense of current and relevant research topics. This has allowed me into the front row seats to witness and participate in the deep learning revolution of the 2010s during which discussions of artificial intelligence and its implications have moved from science fiction novels to the daily news and our cars are beginning to drive themselves.

My colleagues at HIP have been a daily source of inspiration and peer support during my work. Our project leader Assoc. Prof. Mikko Voutilainen has kept a jovial atmosphere in our group as well as made sure we are not lacking in any resources needed for the daily work, whether it is an endless supply of new research questions to look at, friendly banter over a cup of coffee or a top of the line GPU server. Dr. Matti Kortelainen helped me enormously when getting acquainted with the world of tracking particles and gave me a glimpse of what an actual top tier software engineer can do. To Dr. Henning Kirchenmann I am grateful for his friendship during my doctoral work and for launching me on an unexpected but not unwelcome new career trajectory towards Silicon Valley. Teemu Rantalaiho, Dhruv Saksena and Dr. Matias Koskinen, thank you for all the MPS fun both in Finland and in California. To my fellow particle physics doctoral scholars Jaana, Kimmo, Laura, Mikko and Santeri: thank you for sharing the ups and downs of the PhD-life with me.

I wish to thank Prof. Caterina Doglioni for accepting the responsibility of acting as my

opponent in the thesis defence. Additionally my thanks to Dr. Benjamin Nachman and Andreas Salzburger for performing the work of pre-examiners and giving valuable feedback and corrections to the thesis as well as kind comments and encouragement on its quality.

To my friends: There are no words to capture how grateful I am for all of you. We have explored the world together, survived all the courses thrown at us, partied from dusk till dawn and flourished. From exploring the building blocks of the Universe to starting a home brewery and everything in between, I have always found myself in good company. Antti, Natalia and Jussi, thanks for all the galaxies. Anton, Jaska and Kimmo, thank you for all the adventures.

My family has given their unconditional support towards my academic ambitions from the first classes of elementary school all the way through my doctoral studies. While the roots of education are known to be bitter it helps to have them sugar-coated with praise and validation from ones family from an early age. My mother Sari has made sure that I am able to pursuit my interests and dreams throughout my life, whether it has been building miniatures and medieval armour or chasing subatomic particles. In a world of uncertainty it is good to have constants to rely on.

Lissu, out of all the wonderful things that have happened during my studies, you are by far my favourite. I cannot wait to see where we will go next.

Abstract

The LHC particle accelerator at CERN is probing the elementary building blocks of matter at energies never seen in laboratory conditions before. In the process of providing new insights in to the Standard Model describing the current understanding of physics governing the behaviour of particles, the accelerator is challenging the algorithms and techniques used in storing the collected data, rebuilding the collected collision events from the detector signal and analysing the data. For this end many state of the art methods are being developed by the scientist working in the LHC experiments in order to gain as much knowledge from the unique data collected from these particle collisions.

The decade starting from 2010 can be in many respects considered as the deep learning revolution where a family of machine learning algorithms collectively called deep neural networks had significant breakthroughs driven by advances in hardware used to train these algorithms. During this period many achievements previously only seen in the realm of science fiction became reality as the deep neural networks began driving cars, images and videos could be enhanced with super resolution in real time and improvements in automated translation tools lowered the barriers in communication between people. These results have given the field of deep learning a significant momentum and lead to the methods spreading across academic disciplines as well as different industries.

In this thesis the recent advances of deep learning are applied into the realm of particle physics using the data collected by the CMS experiment at the LHC at CERN. First topic presented considers the task of rebuilding the flight paths of charged particles called tracks inside the detector using the measurements made by the Tracker sub-detector in the heart of the CMS. The conditions present inside the detector during particle collisions demand for advanced algorithms able to be both fast and precise. The project in this thesis looks at estimating the quality of the reconstructed tracks and reject tracks that look like they are a result of mistakes made by the reconstruction algorithms, purifying the reconstructed dataset from false signals. Previously the task has been done initially by cut based selections determined by physicists and later by another machine learning algorithm known as the boosted decision tree. Here the first application of deep neural networks to the task is presented with the goal of both simplifying the upkeep of the classifier as well as improving the performance.

In the second topic the application of deep neural network classifiers in the context of a search for a new particle, the charged Higgs boson, is presented. Here the main focus is in producing a classifier that has been decorrelated from a variable of interest that will be used in making the final discovery or exclusion of the hypothetical particle. The classifier can then be used just like any other selection step in the analysis aiming to separate known Standard Model background events from the expected signal without distorting the distribution for the variable of interest.

Both research topics present first time use cases at the CMS for deep neural networks in their respective contexts and the work done includes the full stack of solving a machine learning problem, starting from data collection strategy to cleaning the data and working out the meaningful input variables for the problem all the way to training, optimizing and deploying the models to get the final results for their performance.

Publications

1. CMS Collaboration (including J.Havukainen), "Search for charged Higgs bosons with the $H^{\pm} \to \tau^{\pm}\nu_{\tau}$ decay channel in the fully hadronic final state at $\sqrt{s} = 13$ TeV", *CMS Physics Analysis Summary* CMS-PAS-16-031, 2016 (`https://cds.cern.ch/record/2223865`

2. CMS Collaboration (including J.Havukainen), "Search for charged Higgs bosons with the $H^{\pm} \to \tau^{\pm}\nu_{\tau}$ decay channel in proton-proton collisions at $\sqrt{s} = 13$ TeV", *JHEP* **07** (2019) 142, 2019. (`https://arxiv.org/abs/1903.04560`

Author's contributions

The major contributions of the author are presented in this monograph thesis in part IV CMS Track Classification and part V Search for the Charged Higgs boson in the $\tau^{\pm}\nu_{\tau}$ hadronic decay channel where the development and deployment of deep learning based solutions to track classification and Charged Higgs boson search are presented. These projects have been carried out solely by the author and the results presented here are the author's private work.

The author has also participated in the code base development and preparation of the two publications above in the search of the Charged Higgs boson as a part of CMS experiment group in the Helsinki Institute of Physics.

Additionally the author has participated in the Run 2 data taking of the CMS experiment working in the shift crew as the central DCS shifter, worked on the upkeep and development efforts in the CMS tracking group with regards to the track quality classifiers, performed the task of offline tracking validator at the CMS and done research work for jet energy corrections using deep neural networks in the DeepJet group at the CMS. In the outreach efforts the author has organized and volunteered at the "Hands on particle physics" events organized for high school students by the International Particle Physics Outreach Group.

"Simple. Real stupidity beats artificial intelligence every time."
– Terry Pratchett, *Hogfather*

Contents

Acronyms and symbols

ATLAS	A Toroidal LHC ApparatuS
AUC	Area Under the Curve
BDT	Boosted Decision Tree
BSM	Beyond Standard Model
CA	Cellular Automaton
CERN	The European Center for Nuclear Research
C.L.	Confidence Level
CMS	Compact Muon Solenoid
CMSSW	CMS Software Framework
CSC	Cathode Strip Chamber
CSV	Combined Secondary Vertex
CTF	Combinatorial Track Finder
DNN	Deep Neural Network
DT	Drift Tube
ECAL	Electromagnetic Calorimeter
EWK	Electroweak
FPGA	Field Programmable Gated Array
GAN	Generative Adversarial Network
GMM	Gaussian Mixture Model
GNN	Graph Neural Network
GSF	Gaussian Sum-Filter
HB	Hadron barrel calorimeter
HCAL	Hadronic Calorimeter
HE	Hadron endcap calorimeter
HL-LHC	High-Luminosity Large Hadron Collider
HLT	High Level Trigger
IP	Interaction Parameter
JER	Jet Energy Resolution
JES	Jet Energy Scale
JSD	Jensen-Shannon Divergence
LHC	Large Hadron Collider
LO	Leading order
LSTM	Long-Short Term Memory
L1T	Level 1 Trigger
MAE	Mean Absolute Error
ML	Machine Learning
MVA	Multivariate Analysis
MSE	Mean Squared Error
NLO	Next to leading order
PDF	Parton Distribution Function
PF	Particle Flow
QCD	Quantum Chromodynamics
QED	Quantum Electrodynamics
ROC	Receiver Operating Characteristic

RPC	Resistive Plate Chamber
SGD	Stochastic Gradient Descent
SM	Standard Model (of particle physics)
SUSY	Supersymmetry
TEC	Tracker end cap
TIB	Tracker inner barrel
TOB	Tracker outer barrel
2HDM	Two Higgs Doublet Models

$\mathcal{B}$	Branching fraction
c	speed of light in vacuum
η	pseudorapidity
ϕ	Azimuthal angle
Γ	decay width
$\mathcal{L}$	Lagrangian density
m	Invariant mass
m_{T}	Transverse mass
$\vec{p}_{\mathrm{T}}$	Transverse momentum
$\vec{p}_{\mathrm{T}}^{\,\mathrm{miss}}$	Missing transverse momentum
ΔR	Distance in $\eta - \phi$ plane
σ	Cross section or sigmoid function
τ_{h}	Hadronically decaying tau

Conventions

The CMS coordinate system is described in Section 13.1. In this coordinate system the physics object kinematics are parametrized in terms of $(p_{\mathrm{T}}, \eta, \phi, m)$. This convention is applied throughout the thesis.

Unless otherwise specified natural units where $\hbar = c = 1$ are used in the thesis. In this unit system energy, momentum and mass are related as $E^2 = p^2 + m^2$ so that any of the three variables can be expressed in terms of electron volts (eV). In particle physics usually the energies are in the scale of giga electron volts (GeV = 10^9 eV) or tera electron volts (TeV = 10^12 eV).

The missing transverse momentum $\vec{p}_{\mathrm{T}}^{\,\mathrm{miss}}$ refers to type-I corrected missing transverse momentum unless otherwise stated.

Charge conjugation is implied unless a distinction is made. This means that with "electrons" refer collectively to electrons and positrons. Also in the case of processes a collective term is used to imply both processes i.e. $\mathrm{H}^+ \to \tau^+\nu_\tau$ and $\mathrm{H}^- \to \tau^-\bar{\nu}_\tau$ is denoted as $\mathrm{H}^\pm \to \tau\nu_\tau$.

Part I

The Standard Model of particle physics

Chapter 1

Introduction

The physical world around us consists of elementary building blocks called particles and the interactions between them. The Standard Model describes these particles and interactions in terms of quantum field theory and it is the culmination of decades of work that is described as the crown jewel of particle physics, providing a framework which has served as a foundation for the research of particle physics in both theoretical and experimental directions nearly half a century now. The theory combines the electroweak interaction described by Glashow-Weinberg-Salam model [1–3] discussed in Chapter 4 and quantum chromodynamics (QCD) [4–6] concerning the strong interaction described in Chapter 6. The remaining fundamental force of gravity is not included in the Standard Model, but the search for models unifying general relativity describing gravity and the Standard Model remains an active field of research to this day and it will be described further along with other Beyond Standard Model (BSM) theories in Chapter 7. The latest triumph of the Standard Model was the finding of the Higgs boson in 2012 [7, 8] at the Large Hadron Collider (LHC) at CERN. It provides the explanation to how elementary particles obtain their masses and completes the Standard Model framework. The Higgs boson and the Higgs mechanism are presented in Chapter 5.

Chapter 2

Particles

The Standard Model contains a total of twelve matter particles called *quarks* and *leptons* and five interaction particles referred to as *bosons* that mediate the electroweak and strong interactions between the matter particles. Each of the particles can be described with a set of properties consisting of mass, spin, electric charge and color charge. The particles in the Standard Model are observable excitations of the underlying quantum fields and the fields are the fundamental objects that are used in formulating the mathematical framework for particle physics.

The particle content and some properties of the particles of the Standard Model are presented in schematic 2.1. The following sections briefly detail the different particle categories and their properties.

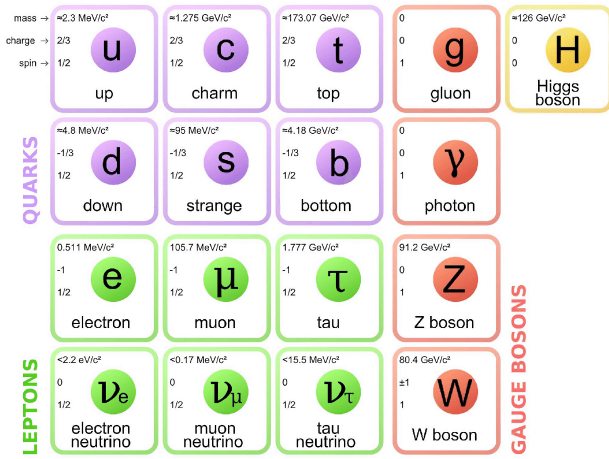


Figure 2.1: Particle content of the Standard Model. Three leftmost columns are the three fermion generations making up all known matter and the remaining two columns contain the interaction mediating bosons.[9]

2.1 Fermions

The fermions in the Standard Model are taken from observations of the physical world. Almost every object encountered in the ordinary conditions for human life is made of only three particles: *up* quark, *down* quark and *electron*. However this is only a quarter of all the known fermions in the Universe. Common to all fermions is that they have a *spin* value of 1/2. Spin can be though of as a quantum mechanical version of angular momentum intrinsic to a particle explaining the name. Spin plays a crucial role in how fermions behave as fermions obey the Pauli exclusion principle stating that no two identical fermions can occupy the same state quantum state. This property of fermions prevents electrons from all falling into the lowest energy state around a nucleus and dictates how electron structures around nuclei are organized.

Quarks are the elementary particles that make up protons and neutrons that are found in the nuclei of atoms. Quarks interact primarily through strong interaction, the force responsible for keeping the nuclei together. The Standard Model contains three generations of quarks split into up-type and down-type quarks. The up-type quarks have an electric charge of $2/3e$ while the down-type quarks have a charge of $-1/3e$. In addition to strong interaction quarks also interact through electromagnetic interaction and weak interaction, enabling phenomena like nuclear fission and fusion powering the Sun and all the other stars. In addition to up and down quarks the rest are in order of ascending mass: *strange*, *charm*, *bottom* and *top*. Quarks carry a *color charge* dictating their interactions through the strong force. The nature of this interaction is such that a color charged object can never be observed but instead all particles with a color charge will only exist in color neutral combinations. This leads to up and down quarks combining into protons

Leptons contain particles like the electron responsible for all things considered chemistry, binding atoms together into molecules making up most of the materials around us. Electrons can be found on their own but due to their electric charge of $-1e$ free electrons are attracted toward particles with positive electric charge and as such mostly the electrons in mundane environments are found around a nucleus. On top of electrons there are two other leptons with an electric charge: *muon* and *tau*. With most respects these two particles are like heavier copies of the electron. However the difference in mass leads to muon and tau being of interest in particle collisions, where muons can often be used as a clean and easy to measure signal and taus are present in many exciting collision events due to the relatively high mass providing a significant coupling to the Higgs boson. There are also leptons without an electric charge called *neutrinos*. They only interact through the short ranged weak interaction, resulting in them being rarely observed in every day environments even though there are myriads of neutrinos passing through us at any given moment. There are three of these near massless particles called electron neutrino, muon neutrino and tau neutrino corresponding to electron, muon and tau respectively. These neutrinos are produced in processes like nuclear fission and fusion.

2.2 Bosons

Physical forces are described as exchange of force mediating particles, bosons, in the Standard Model. Where fermions had a half-integer spin value and followed Fermi-Dirac statistics preventing two fermions from existing in the same quantum state, bosons have an integer spin value and follow Bose-Einstein statistics. This allows any number of bosons to simultaneously occupy the same quantum state. Bosons in the Standard Model appear through symmetries in the Lagrangian describing the fermion content of the model as described in 3.

The boson describing the strong interactions among fermions is the *gluon*. It is a massless particle without an electric charge. It however carries a color charge allowing it to interact with quarks. Although the boson itself is massless, the range of the strong interaction is very small, limited to the scale of an atom nucleus. This is a result of the *color confinement* phenomenon in Quantum Chromodynamics (QCD) describing the strong interaction. It states that color charged objects cannot be observed in isolation, but instead they form color neutral combinations by producing new hadrons to balance out the color charge. A concrete consequence of this is that when two quarks are being pulled apart, the energy required to move the particles further from each other increases with the distance until it is energetically favourable for the quark system to produce a new quark-antiquark pair to form color neutral combinations of both of the original quarks. This process of *hadronization* is the reason why hadron-hadron colliders like the Large Hadron Collider (LHC) at CERN observe a large quantity of objects called *jets*, originating from a quark or gluon and ending up as a spray of color neutral combinations of quarks. Particles formed by quarks bound together into color neutral combinations are called baryons (odd number of quarks) and mesons (even number of quarks).

Photon is the most commonly known boson in layman's vocabulary and it was also the first of the bosons experimentally observed in the early decades of the 20th century. The photon existing as a particle with a quantized energy challenged the wave theories that had been used to explain properties of light until then and demanded that both the wave and particle like characteristics of quantum scale objects to be recognised. In the terms of the Standard Model photon is the boson responsible for mediating the electromagnetic interactions between electrically charged objects. Photons are massless particles and able to mediate the electromagnetic interaction with an infinite range. Sight, electricity and all of chemistry are just results of electrons and protons exchanging photons among each other.

$W^{\pm}$ and Z bosons are the force carriers of the weak interaction. Notably there are three different bosons mediating the weak force, two of them electrically charged and one neutral. The nomenclature weak is used because the effect of weak interaction is typically orders of magnitude smaller than the strong or electromagnetic interaction. The bosons of weak interaction are relatively massive, causing them to decay into lighter particles in a short time scale, restricting the range of the weak interaction to be limited mostly to subatomic scales. The weak interaction is the only force that can change the flavour of a fermion and breaks the parity symmetry setting right- and left-handed particles on a different setting, so that only left-handed fermions and right-handed anti-fermions interact through the weak

interaction. It is the flavour changing ability of the weak interactions that makes it possible for a neutron to decay into a proton through one of it's quarks changing flavour from down quark to up quark. Where the strong interaction affected particles with a color charge and electromagnetic interaction takes place between particles with electric charge, the weak interaction happens between particles with *weak isospin*. Left-handed fermions have a weak isospin of $\pm 1/2$ while right-handed fermions have weak isospin of 0. Weak isospin is conserved in reactions. The massiveness of the interaction mediating bosons is another curious feature in the weak interaction, one that lead to first the theory and half a century later of the most elusive part of the Standard Model of particle physics: the *Higgs boson*.

2.3 Higgs boson

The mass of the $W^{\pm}$ and Z bosons raise an issue in the theory: Introducing the mass term of the bosons to the Lagrangian of the Standard Model breaks the symmetry of the Lagrangian, causing it to be unrenormalizable. Renormalizability is considered necessary in physical theories as renormalization techniques are used to treat infinities otherwise arising in quantum calculations. A way around this issue was found by three different parties around the same time in 1964: Brout and Englert [10], Guralnik, Hagen and Kibble [11] and Higgs [12]. Out of the six Higgs and Englert were awarded the Nobel prize for this work in 2013 [13].

Their idea was to have the Lagrangian itself stay invariant under the symmetry but have the system evolve into a ground state where the symmetry gets broken, earning the name *spontaneous symmetry breaking*. The Goldstone theorem [14] states that if an exact and continuous global symmetry if broken, it will generate a massless scalar particle in the theory. These massless particles are absorbed into the existing gauge bosons as an additional degree of freedom: mass. The number of broken symmetries equals the number of new particles that can be used to bestow mass, and in the case of electroweak symmetry breaking in the Standard Model the number of the broken symmetries is three. This allows masses for W^+, W^- and Z bosons, leaving the photon massless as expected.

In order to produce this spontaneous symmetry breaking effect a new scalar field is added to the Standard Model. On top of producing the Goldstone particles, the new field also introduces a new particle to the Standard Model: The Higgs Boson.

Chapter 3

Quantum Electrodynamics

The interactions between light and matter are described by Quantum Electrodynamics (QED). Hence the particles entering the QED Lagrangian are leptons and photons. For the purposes of this short presentation of the topic quarks are ignored even though they carry an electric charge as well. So the particle content is made out of electrons, muons, taus and the photons mediating the interactions between them. However as mentioned earlier, muons and taus differ from electron only by their mass parameters so that we can simplify the treatment of the Lagrangian to only the electron and photon.

The credit for the development of the theory of interactions between light and matter can be credited to multiple people from the first half of 20th century. Dirac provided the quantum description for interactions of atoms and radiation field [15], Tomonaga presenting the Lorentz invariant formulation [16] and the three papers solving the divergence problems in the theory by Feynman [17], Schwinger [18] and Tati & Tomonaga [19] resulted in a renormalizable theory of quantum electrodynamics.

3.1 Dirac equation

The Dirac equation [20] gives the quantum mechanical description of charged particles with spin 1/2:

$$(i\gamma^\mu \partial_\mu - m)\psi = 0, \tag{3.1}$$

where γ^μ are the Dirac matrices, m is the particle's mass and ψ is a Dirac four-spinor which has a dependency on the coordinates x, that is $\psi = \psi(x)$. The four spinor contains the wave functions for both the particle and it's antiparticle. This equation containing the dynamics that describe the behaviour of leptons is reached by first formulating the Lagrangian describing the particle content of the model and then minimizing the action by applying the Euler-Lagrange equations. The Lagrangian describing the spin 1/2 charged particles is

$$\mathcal{L}_{\text{Dir}} = i\bar{\psi}\gamma^\mu\partial_\mu\psi - m\bar{\psi}\psi = \bar{\psi}(i\not{\partial} - m)\psi, \tag{3.2}$$

using $\bar{\psi} = \gamma^0\psi$ and $\not{\partial} = \gamma^\mu\partial_\mu$. Applying the Euler-Lagrange equation gives equation of motion (3.1):

$$\begin{aligned} &\frac{\partial\mathcal{L}}{\partial\bar{\psi}} - \frac{\partial}{\partial x^\mu}\frac{\partial\mathcal{L}}{\partial(\partial\bar{\psi})} = 0 \\ &\Leftrightarrow (i\gamma^\mu\partial_\mu - m)\psi = 0. \end{aligned} \tag{3.3}$$

This is a standard method for finding the equations of motion for a physical system once the Lagrangian is defined based on some prior assumptions as was done here on the particle content.

3.2 U(1) symmetry

The Dirac Lagrangian (3.2) remains invariant under the U(1) symmetry group containing 1x1 matrices satisfying $\mathbf{U}^*\mathbf{U} = \mathbf{U}\mathbf{U}^* = 1$, that is unitary matrices. This symmetry group is the set of all complex numbers with unit magnitude $\text{U}(1) = \{e^{i\theta}|\theta \in \mathbb{R}\}$.

This invariance means that the Lagrangian describing the leptons can be manipulated with transformations belonging to U(1) without changing it. This can be shown by letting the fields transform as:

$$\begin{aligned} \psi \to \psi' &= e^{-ie\alpha}\psi \\ \bar{\psi} \to \bar{\psi}' &= e^{ie\alpha}\psi. \end{aligned} \tag{3.4}$$

Here the e in the exponential will turn out to signify the elementary electric charge, so it is just a constant. When inserting these to the Lagrangian in Equation (3.2) it becomes evident that it is invariant under this global transformation:

$$\begin{aligned} \mathcal{L}_{\text{Dir}} \to \mathcal{L}'_{\text{Dir}} &= e^{ie\alpha}\bar{\psi}(i\not{\partial} - m)e^{-ie\alpha}\psi \\ &= e^{ie\alpha}e^{-ie\alpha}\bar{\psi}(i\not{\partial} - m)\psi \\ &= \bar{\psi}(i\not{\partial} - m)\psi \\ \Leftrightarrow \mathcal{L}'_{\text{Dir}} &= \mathcal{L}_{\text{Dir}}. \end{aligned} \tag{3.5}$$

A straightforward method to produce the *gauge field theory* describing the lepton and photon and their interactions is by now promoting the found U(1) global symmetry of the Lagrangian into a local symmetry, reintroduce necessary correction terms to maintain the symmetry property and interpret these terms as the interactions between the fermion fields and a gauge field. Starting by allowing the rotation angle α in the transformation to depend on the coordinates x, giving the new transformations:

$$\begin{aligned} \psi \to \psi' &= e^{-ie\alpha(x)}\psi \\ \bar{\psi} \to \bar{\psi}' &= e^{ie\alpha(x)}\psi. \end{aligned} \tag{3.6}$$

Applying these, the Lagrangian becomes

$$\begin{aligned} \mathcal{L}'_{\text{Dir}} &= e^{ie\alpha(x)}\bar{\psi}(i\not{\partial} - m)e^{-ie\alpha(x)}\psi \\ &= e^{ie\alpha(x)}e^{-ie\alpha(x)}\bar{\psi}(i\not{\partial} - m)\psi - \bar{\psi}(i^2 e\gamma^\mu\psi\partial_\mu\alpha(x)) \\ &= \mathcal{L}_{\text{Dir}} + e\bar{\psi}\gamma^\mu\psi\partial_\mu\alpha(x). \end{aligned} \tag{3.7}$$

As a result of allowing the angle to depend on the location x, that is promoting the transformation from global to local, the Lagrangian picks up a new term when the transformation is applied. This means the Lagrangian has to be modified in order for the U(1) symmetry to be retained, which can be done by introducting a new gauge field A^μ and redefining the partial derivative as the *covariant derivative*:

$$\partial_\mu \to D_\mu = \partial_\mu + ieA_\mu. \tag{3.8}$$

This new derivative will cancel out the excess term in Equation (3.7) if the gauge field is required to transform as

$$A_\mu \to A'_\mu = A_\mu + \partial_\mu\alpha(x) \tag{3.9}$$

under the local U(1) transformations. Now if the partial derivative in (3.2) is replaced with the covariant derivative from (3.8) and the local U(1) transformation is applied to it:

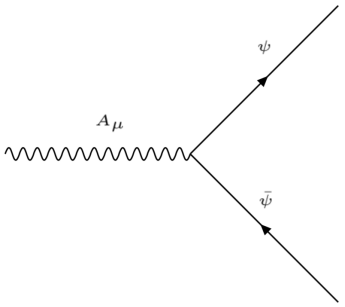


Figure 3.1: The interaction term $\mathcal{L}_{\text{int}} = e\bar{\psi}\gamma^\mu\psi A_\mu$ represented as a Feynman graph. The constant e describes the strength of the interaction.

$$\begin{aligned}\mathcal{L}'_{\text{Dir}} &= \mathcal{L}_{\text{Dir}} + e\bar{\psi}\gamma^\mu\psi\partial_\mu\alpha(x) - e\bar{\psi}\gamma^\mu\psi\partial_\mu\alpha(x) - e\bar{\psi}\gamma^\mu\psi A_\mu \\ &= \mathcal{L}_{\text{Dir}} - e\bar{\psi}\gamma^\mu\psi A_\mu \\ &= \mathcal{L}_{\text{Dir}} + \mathcal{L}_{\text{int}}.\end{aligned} \tag{3.10}$$

This new addition $\mathcal{L}_{\text{int}} = -e\bar{\psi}\gamma^\mu\psi A_\mu$ to the Lagrangian ensures the U(1) symmetry at the local level. It is interpreted as an interaction between the three fields ψ, $\bar{\psi}$ and A_μ. Diagrammatically this is represented as a *Feynman diagram*, as shown in Figure 3.1.

3.3 Identifying the gauge field A_μ

While the Lagrangian now has regained it's invariance also against local transformations of U(1), the new gauge field A_μ still needs to be identified. Currently it is known how it behaves under U(1) transformations and that it is a *vector field*, having both a direction and magnitude at each point x. Since the aim is to produce a theory about interactions between matter and light, the electromagnetic four-potential is a candidate for the role of this gauge field: $A^\mu = (\frac{\phi}{c}, \vec{A})$, where ϕ is the scalar electric potential and $\vec{A}$ is the vector magnetic potential. Using this four-potential, the electric and magnetic field can be written as

$$\begin{aligned}\vec{E} &= -\Delta\phi - \frac{\partial\vec{A}}{\partial t} \\ \vec{B} &= \Delta \times \vec{A}.\end{aligned} \tag{3.11}$$

The components of these fields can be used to define an antisymmetric tensor $F^{\mu\nu}$ as

$$F^{\mu\nu} \equiv \partial^\mu A^\nu - \partial^\nu A^\mu = \begin{bmatrix} 0 & -E_x & E_y & E_z \\ E_x & 0 & -B_z & B_y \\ E_y & B_z & 0 & -B_z \\ E_z & -B_y & B_x & 0 \end{bmatrix} \tag{3.12}$$

Using this tensor the equations of motions describing the electromagnetic fields with sources J^ν called the Maxwell's equations can be written as:

$$\partial_\mu F^{\mu\nu} = J^\nu, \tag{3.13}$$

where $J^\nu = (\rho, \vec{J})$ is the four-current with charge density ρ and current density $\vec{J}$. These equations of motion are derived from a Lagrangian

$$\mathcal{L}_\gamma = -\frac{1}{4} F^{\mu\nu} F_{\mu\nu}. \tag{3.14}$$

in a similar manner as the Dirac equation was derived above. As an inner product of two tensors this Lagrangian is Lorentz invariant and it is also invariant under the transformation imposed earlier for the gauge field in Equation 3.9 which can be seen by explicitly writing open the tensors as

$$\begin{aligned} \mathcal{L} \to \mathcal{L}' &= -\frac{1}{4} F'^{\mu\nu} F'_{\mu\nu} \\ &= -\frac{1}{4} (\partial^\mu A'^\nu - \partial^\nu A'^\mu)(\partial_\mu A'_\nu - \partial_\nu A'_\mu) \\ &= -\frac{1}{4} (\partial^\mu [A^\nu + \partial^\nu \alpha(x)] - \partial^\nu [A^\mu + \partial^\mu \alpha(x)])(\partial_\mu [A_\nu + \partial_n \alpha(x)] - \partial_\nu [A_\mu + \partial_\mu \alpha(x)]) \\ &= -\frac{1}{4} (\partial^\mu A^\nu - \partial^\nu A^\mu)(\partial_\mu A_\nu - \partial_\nu A_\mu) - \frac{1}{4} (\partial^\mu \partial^\nu \alpha(x) - \partial^\nu \partial^\mu \alpha(x))(\partial_\mu \partial_\nu \alpha(x) - \partial_\nu \partial_\mu \alpha(x)) \\ &= -\frac{1}{4} F^{\mu\nu} F_{\mu\nu} \\ \Leftrightarrow \mathcal{L}' &= \mathcal{L}. \end{aligned} \tag{3.15}$$

This demonstrates that the Lagrangian and the electromagnetic four-potential used in describing photons has the required transformation qualities as the new gauge field needed in

the Dirac equation to preserve the invariance in the local U(1) transformations. Since QED is supposed to be the theory of light, matter and their interactions, the suggestively named gauge field A^μ added to the Dirac Lagrangian should be interpreted as the electromagnetic four-potential and the additional kinetic term $\mathcal{L}_\gamma$ should be added to the full Lagrangian in order to allow the photon equations of motions to be also derived from the Lagrangian. This gives the Lagrangian the form:

$$\begin{aligned}\mathcal{L}_{\text{QED}} &= \mathcal{L}_{\text{Dir}} + \mathcal{L}_\gamma + \mathcal{L}_{\text{int}} \\ &= \bar{\psi}(i\not{\partial} - m)\psi - \frac{1}{4}F^{\mu\nu}F_{\mu\nu} - e\bar{\psi}\gamma^\mu A_\mu \psi.\end{aligned} \tag{3.16}$$

As noted earlier the leptons share other relevant properties except their mass, so the Lagrangian is trivial to extend to include the three lepton flavours, electrons, muons and taus. This similarity between leptons is known as the *lepton universality* and it has been shown to hold within experimental accuracy in multiple tests, although recent results display some tensions challenging this assumption [21].

$$\mathcal{L}_{\text{QED}} = \sum_{i=1}^{3}(\bar{\psi}_i(i\not{\partial} - m_i)\psi_i - e\bar{\psi}_i\gamma^\mu A_\mu \psi_i) - \frac{1}{4}F^{\mu\nu}F_{\mu\nu} \tag{3.17}$$

with $i = 1, 2, 3$ representing electron, muon and tau. This is the description of the interactions between leptons and light, where two electrons repel each other by exchanging photons.

Quantum electrodynamics was the first portion of the Standard Model that was formulated. What was presented here was only the fundamentals of the theory as there is some finesse required to take care of the edge cases, which is emphasised by the fact that the three papers by Feynman, Schwinger and Tati & Tomonaga mentioned at the beginning of this chapter were awarded Nobel prizes for their solutions to the divergence problems of the theory. However this demonstrated the important approach of how the interactions between matter and force mediating gauge bosons are introduced into the theory by first including non-interacting fields describing the matter content, finding a global continuous invariance they satisfy, promoting the invariance into a local one and adding a gauge field to ensure the invariance is satisfied by the Lagrangian. The same approach can be used with the two remaining forces, the weak interaction described by Quantum Flavourdynamics and the strong interaction described by Quantum Chromodynamics. However the derivation there is a bit more demanding since the symmetry groups corresponding to them are not as simple as the U(1) group.

Chapter 4

Quantum Flavourdynamics

The weak interaction is responsible for all flavour changing interactions in the Standard Model and as such enables reactions that transform nuclei in nuclear reactions. It affects all the fermions in the Standard Model, that is all of them have a coupling to gauge fields responsible for the weak interaction mediating bosons $W^{\pm}$ and Z. At low energies this interaction almost seems like a contact force as the high masses of the gauge bosons cause them to decay quickly, restricting the interaction to short distances. This is one of the reasons why the first model trying to explain the β -decay by Enrico Fermi in 1933 modelled the phenomenon as a four-particle interaction. [22]

4.1 Parity violation

Violation of parity conservation is another feature of the weak interaction. Parity is the inversion of a coordinate in a system with a parity operator $\mathcal{P}$ so that a wave function describing a particle would transform as $\mathcal{P}\phi(x) = \phi(-x)$. This violation of parity introduces an inequality between *left-handed* and *right-handed* particles in the Standard Model where handedness refers to a property of every non-integer spin particle called *chirality*. The parity violation in weak interactions was first put forth by Lee and Yang in 1956 [23].

Chirality can be defined by how a particle's wave function behaves when operated on by the Dirac gamma matrix γ^5:

$$\begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \tag{4.1}$$

Fermion wave functions have eigenvalues of ± 1 under this operator corresponding to right- and left-handed particles. This parity violation manifests itself in only left-handed particles

and right-handed antiparticles interacting with the $W^{\pm}$ gauge bosons mediating the *charged current* weak interaction. Interactions involving Z boson are similarly referred to as the *neutral current* weak interactions as Z boson is electrically neutral and the interactions do not change the electric charges of the interacting particles unlike in the charged current case.

Charge-parity conservation is also violated by the weak interaction which results in different behaviour of particles and antiparticles in the Standard Model. This type of violation is important to be included in the model as it is required for feasible explanations as to why the observable universe contains so much less antimatter than matter, a problem referred to as the baryon asymmetry. The charge-parity violation was first observed in decays of K^0 mesons by Christenson *et al.* in 1964 [24].

4.2 SU(2) symmetry group

For the theory of particle physics to contain these features, the fields used to describe fermions need to be split by chirality. This will have the effect of making the theory *non-Abelian* meaning that the order of gauge transformations of the symmetry group will matter, unlike in the case of U(1) symmetry in QED which was an Abelian theory. The non-Abelian nature of the symmetry operators will result in the subsequent gauge bosons to interacting among themselves as will be demonstrated below. Separating the fields into two fields with left- and right-handedness can be done in a straightforward manner using the γ^5 operator introduced above in Eq. (4.1):

$$\psi = \psi_L + \psi_R \tag{4.2}$$

where

$$\begin{aligned} \psi_L &= P_L\psi = \frac{1-\gamma^5}{2}\psi \\ \psi_L &= P_L\psi = \frac{1+\gamma^5}{2}\psi. \end{aligned} \tag{4.3}$$

P_L and P_R are known as projection operators and they pick two orthogonal components out of the original field corresponding to the left and right chirality states. These operators satisfy the requirements

$$P_R P_R = P_R, \quad P_L P_L = P_L, \quad P_L P_R = P_R P_L = 0. \tag{4.4}$$

With these the lepton Lagrangian containing electrons, muons and taus can be written with left- and right-handed components

$$\begin{aligned}\mathcal{L}_{\text{leptons}} &= \mathcal{L}_{L,\text{leptons}} + \mathcal{L}_{R,\text{leptons}} \\ &= \sum_{i=1}^{3} (\bar{\psi}_i^L (i\not{\partial} - m)\psi_i^L + \bar{\psi}_i^R (i\not{\partial} - m)\psi_i^R)\end{aligned} \tag{4.5}$$

with $i = 1, 2, 3$ corresponding to electrons, muons and taus. In order to include the left-handed neutrinos (right-handed antineutrinos) that interact through the weak interaction to the Lagrangian, the left-handed fields are arranged into doublets with an electrically charged and a neutral field while the right-handed fields are kept as singlets of charged lepton fields:

$$\psi_1^L = \begin{pmatrix} \nu_e \\ e \end{pmatrix}, \quad \psi_2^L = \begin{pmatrix} \nu_\mu \\ \mu \end{pmatrix}, \quad \psi_3^L = \begin{pmatrix} \nu_\tau \\ \tau \end{pmatrix}, \quad \psi_1^R = e, \quad \psi_2^R = \mu, \quad \psi_3^R = \tau \tag{4.6}$$

The upper and lower field in the doublets differ have an electric charge difference of one. This is required as they will form vertices with the charged weak current carrier $W^\pm$ which will carry off one unit of electrical charge from the vertex. Similarly as was done with QED, the Lagrangian for this theory of weak interactions in Equation (4.5) is found to be invariant under continuous global transformations of the form

$$\psi_i^{L/R} \to \mathbf{U}\psi_i^{L/R}, \quad \bar{\psi}_i^{L/R} \to \bar{\psi}_i^{L/R}\mathbf{U}^\dagger \tag{4.7}$$

with $\mathbf{U}$ being 2x2 matrices satisfying

$$\mathbf{U}\mathbf{U}^\dagger = \mathbf{U}^\dagger\mathbf{U} = \mathbb{1}, \quad \det(\mathbf{U}) = 1. \tag{4.8}$$

This group is called the special unitary group of order two or SU(2) in short. The transformations by this group can be represented with three linearly independent generators and the Lagrangian is invariant with respect to all three. This means that when promoted to local invariances, they will introduce three new gauge fields corresponding to three new bosons in the theory. Similarly as was done with the QED case, lepton universality is assumed here

and the process is demonstrated with respect to just one of the leptons to keep the equations simple. The three generators T^i are

$$T^1 = \frac{1}{2}\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad T^2 = \frac{1}{2}\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad T^3 = \frac{1}{2}\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \tag{4.9}$$

satisfying a commutation relation

$$[T^i, T^j] = \epsilon_{ijk} T^k. \tag{4.10}$$

Using the generators and real number parameters ω_i, the transformations $\mathbf{U}$ of the SU(2) group can be written as

$$\mathbf{U} = e^{i\omega_i T^i}. \tag{4.11}$$

Next allowing the transformations to depend on the coordinate x. The SU(2) transformations affect only the left-handed component of the fields but in the interest of clarity, in the following equations the notation is simplified by writing $\psi_i^L \rightarrow \psi_i$ and similarly for the complex conjugate fields, and the right-handed component of the fields is handled separately at the end of the chapter. The local transformation on the Lagrangian

$$\mathcal{L} \rightarrow \mathcal{L}' = \bar{\psi}_i U^\dagger(x)(i\not{\partial} - m_i)U(x)\psi_i \quad = \bar{\psi}_i(i\not{\partial} - m_i)\psi_i + i\bar{\psi}_i U^\dagger(x)(\not{\partial}U(x))\psi_i) \tag{4.12}$$

In order to remove the additional term picked up by the partial derivative operating on the transformation operator, new gauge fields $W_\mu = W_\mu^i T_i$ with $i = 1, 2, 3$ are introduced to through the covariant derivative:

$$D_\mu = \partial_\mu \mathbb{1} + igW_\mu. \tag{4.13}$$

Here g a constant describing the coupling strength of these new gauge fields. In order to satisfy the local invariance, the gauge fields W_μ must transform as

$$W_\mu \to U(x) W_\mu U^\dagger(x) + \frac{i}{g}(\partial_\mu U(x)) U^\dagger(x). \tag{4.14}$$

This can be explicitly demonstrated by exchanging the partial derivative with the covariant derivative and applying the SU(2) transformation to the fields of the Lagrangian:

$$\begin{aligned}
\mathcal{L} &= \bar{\psi}(i\not{D} - m_i)\psi_i \\
\mathcal{L}' &= \bar{\psi}_i(i\not{\partial} - m_i)\psi_i + i\bar{\psi}U^\dagger(\not{\partial}U)\psi_i - i\bar{\psi}_i U^\dagger U U^\dagger(\not{\partial}U)U^\dagger U\psi_i - g\bar{\psi}_i U^\dagger U\gamma^\mu W_\mu^i U^\dagger U\psi_i \\
&= \bar{\psi}_i(i\gamma^\mu(\partial + igW_\mu) - m_i)\psi_i + i\bar{\psi}_i U^\dagger(\not{\partial}U)\psi_i - i\bar{\psi}_i U^\dagger(\not{\partial}U)\psi_i \\
&= \bar{\psi}_i(i\not{D} - m_i)\psi_i \\
\Leftrightarrow \mathcal{L} &= \mathcal{L}'.
\end{aligned} \tag{4.15}$$

Using the generators in Equation (4.9) the covariant derivative can be written in terms of the three new gauge fields explicitly

$$D_\mu = \partial_\mu \mathbb{1} + igW_\mu^i T_i = \partial_\mu \mathbb{1} + \frac{ig}{2}\begin{pmatrix} W^3 & W^1 - iW^2 \\ W^1 + iW^2 & -W^3 \end{pmatrix}. \tag{4.16}$$

4.3 The electroweak unification

The components on the off-diagonal end up describing the charged weak interaction. The diagonal terms on the other hand still require another component for the derivation to be completed. The lepton Lagrangian still has the U(1) symmetry that was promoted into a local symmetry that provided the photons in QED. It can also be included into the symmetry group by performing SU(2)×U(1) transformation instead of just SU(2). This transformation is

$$U = e^{i\omega_a T^a} e^{i\alpha Y} \tag{4.17}$$

where Y is the charge describing the interaction resulting from gauge field of the U(1) symmetry. Now allowing all the constants W_a and Y to depend on coordinate x and finding the covariant derivative to counter the additional terms as before:

$$\begin{aligned}\mathcal{L} \to \mathcal{L}' &= \bar{\psi}U^{\dagger}(x)(i\not{\partial} - m)U(x)\psi \\ &= \bar{\psi}(i\not{\partial} - m)\psi + i\omega_a\bar{\psi}(\not{\partial}T^a(x))\psi + iY\bar{\psi}(\not{\partial}\alpha(x))\psi,\end{aligned} \tag{4.18}$$

leading to the covariant derivative of

$$D_\mu = \partial_\mu \mathbb{1} + igW_\mu + ig'\frac{Y}{2}B_\mu. \tag{4.19}$$

Here g and g' are called coupling constants and they end up determining how strongly the gauge fields and matter fields are coupled, affecting how likely certain interactions are. Under the SU(2)×U(1) operation the gauge fields transform as

$$\begin{aligned}W_\mu \to W'_\mu &= e^{i\omega_a T^a(x)}W_\mu e^{-i\omega_a T^a(x)} + \frac{i}{g}(\partial_\mu e^{i\omega_a T^a(x)})e^{-i\omega_a T^a(x)}, \\ B_\mu \to B'_\mu &= B_\mu + \frac{iY}{g'}\partial_\mu\alpha(x).\end{aligned} \tag{4.20}$$

Rewriting the Lagrangian with this SU(2)×U(1) covariant derivative gives

$$\mathcal{L} = \bar{\psi}(i\not{\partial} - m)\psi + ig\bar{\psi}\gamma^\mu W^a_\mu T_a\psi + ig'\bar{\psi}YB_\mu\psi. \tag{4.21}$$

Using the representations for the generators the covariant derivative can now also be written explicitly

$$\begin{aligned}&D_\mu = \partial_\mu\mathbb{1} + igW^a_\mu T_a + ig'\frac{Y}{2}B_\mu\mathbb{1} \\ \Leftrightarrow\ &D_\mu = \begin{pmatrix}\partial_\mu + \frac{ig}{2}W^3_\mu + ig'\frac{Y}{2}B_\mu & \frac{ig}{2}(W^1_\mu - iW^2_\mu) \\ \frac{ig}{2}(W^1_\mu + iW^2_\mu) & \partial_\mu - \frac{ig}{2}W^3_\mu + ig'\frac{Y}{2}B_\mu\end{pmatrix}.\end{aligned} \tag{4.22}$$

Here the off-diagonal terms are identified as describing the charged weak currents and the diagonal terms describe the neutral current. The charge Y determining if the field has

interactions with the U(1) gauge boson is called the *weak hypercharge* and it is defined as

$$Y = 2(Q - T^3), \tag{4.23}$$

where Q is the electric charge and T^3 represents the third component of the *weak isospin*. Weak isospin is a quantum number related to the weak interactions. The left-handed doublets and right-handed singlets of matter fields were organised so that the fields in the doublet differ by one unit in the third component of the weak isospin while the singlets have a weak isospin of zero, leaving them out of the charged current weak interaction. The weak isospin is a conserved quantity in weak interactions and the $W^{\pm}$ boson has a weak isospin of ± 1, so the doublet structure is required to prevent the interaction vertices from breaking the conservation. The right handed singlets have a non-zero weak hypercharge however, leaving them free to interact through the neutral weak current except for the right-handed neutrinos whose electric charge is zero as well as their weak hypercharge. This leaves them to be so called *sterile neutrinos* in the Standard Model as there is no interaction excluding gravity through which they could interact.

Chapter 5

Symmetry breaking

The weak interaction and its gauge bosons posed a problem in the theoretical structure of the theory. In order to retain the gauge invariance the Lagrangian would not permit terms representing the gauge boson mass. The issue is solved by introducing a spontaneous breaking of a symmetry so that the Lagrangian is symmetric but the potential evolves with time so that the ground state of the system ends up being non-symmetric.

The idea for spontaneous symmetry breaking in the Standard Model came from research in superconductors done in 1950's by Bardeen, Cooper and Schrieffer [25]. Nambu and Jona-Lasinio used this method in a simplified model to give masses to particles [26, 27]. The implementation in the Standard Model framework was famously done by three different groups around the same time: Guralnik, Hagen and Kibble [11], Englert and Brout [10] and Higgs [12], and it became later known as the Higgs mechanism.

5.1 The sombrero potential

Continuing with the $\mathrm{SU}(2)_\mathrm{L}\times\mathrm{U}(1)_Y$ transformation discussed in the context of the electroweak interaction. As was shown the Lagrangian can remain invariant with respect to these types of operations so that $U(x)\mathcal{L} = \mathcal{L}$, but the Higgs mechanism relies on having the ground state of the system $|0\rangle$ break this symmetry so that $U(x)\,|0\rangle \neq |0\rangle$. If an exact and continuous global symmetry is broken a massless particle called the Goldstone boson is produced [14]. These additional bosons provide another degree of freedom that will be used to give mass to the physical $W^\pm$ and Z bosons. The symmetry will be broken so that

$$\mathrm{SU}(2)_\mathrm{L} \times \mathrm{U}(1)_Y \to \mathrm{U}(1)_\mathrm{em}. \tag{5.1}$$

That is to say there is still a U(1) symmetry left in the system after the symmetry breaking. The number of broken symmetries will equal the number of new Goldstone bosons that can

be used meaning that this operation will produce three bosons since the SU(2) group had three separate symmetries and U(1) group had one.

In order to modify the Lagrangian so that the symmetry can be spontaneously broken a new field needs to be added. Let the new field be a scalar doublet

$$\Phi \equiv \begin{pmatrix} \phi_0 \\ \phi^+ \end{pmatrix}, \tag{5.2}$$

where ϕ_0 and ϕ^+ are complex scalar fields. The Lagrangian component for this field is

$$\mathcal{L}_{\text{scalar}} = \partial_\nu \Phi^\dagger \partial^\nu \Phi - V(\Phi^\dagger \Phi), \tag{5.3}$$

with the potential term $V(\Phi^\dagger \Phi)$ having the form

$$V(\Phi^\dagger \Phi) = \mu^2 \Phi^\dagger \Phi + \lambda \Phi^\dagger \Phi. \tag{5.4}$$

Here λ and μ are constants defining the shape of the potential. The extrema of the potential can be found from the derivative

$$\frac{\partial V(\Phi^\dagger \Phi)}{\partial \Phi^\dagger} = (\mu^2 + 2\lambda \Phi^\dagger \Phi)\Phi = 0. \tag{5.5}$$

In order to have a potential with a well defined ground state it should be bounded from below, which requires $\lambda \geq 0$. If $\mu^2 > 0$ the shape of the potential ends up having only a single minimum at $\Phi = 0$. If $\mu^2 < 0$ there are minima at $\Phi^\dagger \Phi = -\frac{\mu^2}{2\lambda}$ and a maximum at $\Phi = 0$. The latter leads to the famous sombrero shaped potential. The two possibilities are shown in Figure 5.1.

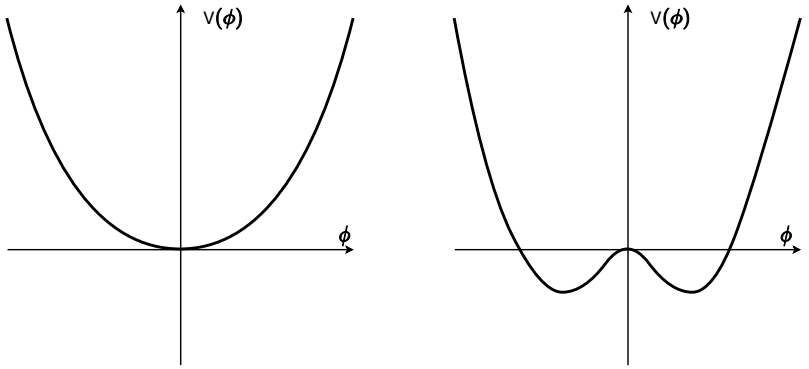


Figure 5.1: The shape of the complex scalar field potential. **Left**: $\mu^2 \geq 0$ leads to a single minimum at $\Phi = 0$. **Right**: $\mu^2 < 0$ leads to a maximum at $\Phi = 0$ and an infinite number of minima at $\Phi^\dagger\Phi = -\frac{\mu^2}{2\lambda}$.

5.2 Breaking the gauge symmetry

The infinite number of minima in the potential is a manifestation of *gauge freedom* for the field Φ. By selecting a gauge to work with a single minimum is chosen which is the ground state of the system. The so called unitary gauge where the massless Goldstone bosons from breaking the symmetry do not appear but instead become the mass terms of the bosons in the Lagrangian is chosen by setting

$$\begin{aligned} \Phi^\dagger\Phi = \phi^{+\dagger}\phi^+\phi_0^\dagger\phi_0 = -\frac{\mu^2}{2\lambda} \rightarrow \phi^+ = 0, \quad \phi_0 = \sqrt{-\frac{\mu^2}{2\lambda}} = v/\sqrt{2} \\ \Leftrightarrow \Phi = \begin{pmatrix} v/\sqrt{2} \\ 0 \end{pmatrix} \end{aligned} . \tag{5.6}$$

Here v is called the *vacuum expectation value* since it is the non-zero constant determining the ground state of the complex scalar field in the Standard Model. Setting the origin at the minimum and expanding the field by perturbing it gives

$$\Phi = \begin{pmatrix} v/\sqrt{2} + H \\ 0 \end{pmatrix}, \tag{5.7}$$

where $H << 1$ and it is allowed to depend on the coordinates $H = H(x)$. $H(x)$ is what is referred to as the *Higgs field*, and after the choice of minimum the original $\mathrm{SU(2)_L \times U(1)}_Y$ is

broken. The doublet Φ is still invariant under transformations of the form

$$\mathbf{U} = e^{-i\frac{\theta}{2}} \begin{pmatrix} e^{i\frac{\theta}{2}} & 0 \\ 0 & e^{-i\frac{\theta}{2}} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & e^{-i\theta} \end{pmatrix}, \quad \mathbf{U}\Phi = \Phi. \tag{5.8}$$

These transformations belong to a representation of the U(1) group and this remaining symmetry is what ends up accounting for the photon to stay massless while the other bosons in the electroweak theory attain mass.

5.3 Mass terms of the Lagrangian

Applying the $\mathrm{SU(2)_L \times U(1)}_Y$ covariant derivative from Eq. 4.22 on this new field to find out what new terms it will produce in the Lagrangian. The scalar Lagrangian becomes

$$\begin{aligned}
\mathcal{L}_{\text{scalar}} = & \frac{1}{2}\partial_\mu H \partial^\mu H + \frac{1}{2}\mu^2 H^2 \\
& + \frac{g^2 v^2}{8}(W^1_\mu W^{1,\mu} + W^2_\mu W^{2,\mu}) \\
& + \frac{v^2}{8}(g' B_\mu - g W^{3,\mu})(g' B^\mu - g W^{3,\mu}) \\
& + \frac{g^2 v}{4}(W^1_\mu + i W^3_\mu)(W^{1,\mu} - i W^{3,\mu}) H + \frac{g^2}{8}(W^1_\mu + i W^3_\mu)(W^{1,\mu} - i W^{3,\mu}) H^2 \\
& + \frac{(g^2 + g'^2)v}{4}(-g W^3_\mu + g' B_\mu)(-g W^{3,\mu} + g' B^\mu) H \\
& + \frac{(g^2 + g'^2)v^2}{8}(-g W^3_\mu + g' B_\mu)(-g W^{3,\mu} + g' B^\mu) H^2.
\end{aligned} \tag{5.9}$$

This expression can be cleaned up by using the definitions

$$W^\pm_\mu = \frac{1}{\sqrt{2}}(W^1_\mu \mp i W^2_\mu), \quad Z^0_\mu = \frac{1}{\sqrt{g^2 + g'^2}}(g W^3_\mu - g' B_\mu), \quad A_\mu = \frac{1}{\sqrt{g^2 + g'^2}}(g W^3_\mu + g' B_\mu), \tag{5.10}$$

where the physical fields corresponding to the bosons are interpreted as mixtures of the gauge fields. This gives the Lagrangian in the form

$$\begin{aligned}\mathcal{L}_{\text{scalar}} = &\frac{1}{2}\partial_\mu H \partial^\mu H \\ &+ \frac{1}{2}\mu^2 H^2 + \frac{g^2 v^2}{4} W_\mu^+ W^{-,\mu} + \frac{(g^2+g'^2)v^2}{8} Z_\mu^0 Z^{0,\mu} \\ &+ \frac{g^2 v}{2} W_\mu^- W^{+,\mu} H + \frac{g^2}{4} W_\mu^- W^{+,\mu} H^2 \\ &+ \frac{(g^2+g'^2)^2 v}{4} Z_\mu^0 Z^{0,\mu} H + \frac{(g^2+g'^2)^2 v^2}{8} Z_\mu^0 Z^{0,\mu} H^2.\end{aligned} \tag{5.11}$$

In this form the scalar Lagrangian has been arranged so that the first row is the dynamic term of the Higgs field, second row contains the mass terms for the three gauge bosons and the last two rows contain the three and four particle interaction terms between the weak gauge fields and the Higgs field. From the mass terms the physical masses of the bosons are identified as

$$m_{W^\pm}^2 = \frac{g^2 v^2}{4}, \quad m_Z^2 = \frac{(g^2+g'^2)v^2}{4}, \quad m_H^2 = 2\mu^2 = 2\lambda v^2 \tag{5.12}$$

Inserting these into the Lagrangian

$$\begin{aligned}\mathcal{L}_{\text{scalar}} = &\frac{1}{2}\partial_\mu H \partial^\mu H \\ &+ \frac{m_H^2}{4} H^2 + m_{W^\pm}^2 W_\mu^+ W^{-,\mu} + \frac{m_Z^2}{4} Z_\mu^0 Z^{0,\mu} \\ &+ \frac{2m_{W^\pm}^2}{v} W_\mu^- W^{+,\mu} H + \frac{m_{W^\pm}^2}{v^2} W_\mu^- W^{+,\mu} H^2 \\ &+ \frac{m_Z^2}{v} Z_\mu^0 Z^{0,\mu} H + \frac{m_Z^2}{2} Z_\mu^0 Z^{0,\mu} H^2.\end{aligned} \tag{5.13}$$

Here an important feature of the interactions between the Higgs field and the other fields is visible in the interaction terms: The couplings are proportional to the mass of the particle. Additionally the gauge field A_μ is not present in the $\mathcal{L}_{\text{Higgs}}$. This is a result of the choice of gauge done when choosing which minimum is the ground state and corresponds to the experimental observation that the photon is a massless particle and as such it does not get a mass term from the Higgs mechanism.

Chapter 6

Quantum Chromodynamics

The last interaction to be included in the Standard Model of particle physics is the strong interaction between particles carrying a *color charge*. The terms of the interactions would be found in a similar manner as was done above for the electric and weak interactions: starting from the Lagrangian describing the quark fields that carry the color charge and finding it to be symmetrical under the global transformations from the SU(3) symmetry group. The theory of strong interaction is called the *quantum chromodynamics* (QCD) and the color charges are assigned to be red, green and blue.

The strong interaction and the quarks were a difficult puzzle in experimental particle physics as new hadrons were being found but quarks were nowhere to be seen at least individually. Deep inelastic scattering experiments between electrons and protons could confirm the existence of inner structure inside the proton [28, 29] that could be explained with new elementary particles inside the proton, but these particles could not be coerced to break free from the proton in order to be observed individually.

The breakthrough discovery for the theory of QCD came with the concept of *asymptotic freedom* by Wilczek and Gross [30] and Politzer [31]. Their key idea was that the strong interaction grows weaker at shorter distances which corresponds to higher collision energies in particle colliders. This allowed QCD to be used in theoretical calculations that could be used to give verifiable experimental results.

6.1 Color carrying gluons

The mediating bosons of the strong interaction are called *gluons*. As was the case for the other interactions the number of massless bosons that get included in the theory by gauging the symmetry reflects the number of generators in the symmetry group. SU(3) can be represented with eight generating 3×3 matrices known as the *Gell-Mann matrices*. The corresponding eight gluons are differentiated by the color combinations they are carrying. This is a notable difference to the other forces covered earlier, since the photon does not carry an electric charge that it would deliver from a fermion to another fermion nor do the Z or $W^{\pm}$ carry the weak charge. Gluons carry a color charge and indeed the exchange of a gluon between

two quarks will alter the color charges of the participating quarks.

The important result of gluons being color charged is that they can interact among themselves as well. The gluon self-interaction leads to the potential between color charged particles to increase as a function of distance so that when two quarks are pulled apart, the force required keeps increasing as the distance grows until there is enough potential energy between the quark pair to produce another quark-antiquark pair as it becomes energetically beneficial. This is known as the *color confinement* of the color charged particles. They are locked into colorless combinations that can be observed like the proton or the neutron and an individual quark with its color charge cannot be observed in the accessible energy scale.

6.2 Hadronization and jets

This color confinement phenomenon results in one of the distinctive experimental features in hadron-hadron colliders used to study the Standard Model: Collimated sprays of particles appear in high-energy interactions between hadrons. When two hadrons hit head-on the Standard Model description of the event is that the constituents of the participating hadrons i.e. quarks exchange a boson transferring momentum between the particles. This exchange of momentum can change the trajectory of some of the quarks in a way that they get pulled away and break the original hadron.

As mentioned above the potential between the quarks keeps increasing with the distance until there is enough potential energy to create another quark-antiquark pair in between the original quarks. However the quark drifting away might still carry enough momentum that it will drift away from its new color charged partner as well creating yet another pair of quarks. This process keeps going until the original quark launched away from the hadron no longer has enough momentum for creating more particles and its locked back into a colorless hadron. This process is known as *hadronization*.

Since the original quark had a significant amount of momentum towards some direction the resulting collection of hadrons have to collectively carry this momentum since it is a conserved quality in physics. This means that this shower of hadrons is moving towards more or less the same direction often at high velocities. This formation process explains the collimated shape of these sprays of hadrons known as *jets*.

These hadronic jets are the observable evidence of quark-quark interactions having taken place in hadron-hadron collisions like the ones taking place at the LHC. They pose a unique experimental challenge since reconstructing the original interaction requires accurately measuring all the particles in the jet which will be discussed more later when considering the reconstruction of physics objects inside particle detectors.

6.3 Sea of quarks and gluons

Although a hadron like the proton is said to be made out of two up quarks and a down quark around these *valence quarks* there are *sea quarks* around the nucleus popping in and out of existance through creation and annihilation of quark-antiquark pairs out of the vacuum and

the gluons that are being exchanged among the quarks. The momentum that is assigned to a hadron is in fact distributed between all of these constituent particles instead of belonging just to the valence quarks.

This complicated structure leads to another experimental feature in hadron-hadron collisions: While the collisions are said to be taking place in some definite energy like 14 TeV the two elementary particles exchanging bosons with each other will never have that exact total energy. Instead the collision energy refers to the center-of-mass energy of the hadron-hadron system and the interaction energy depends on which portion of the total momentum of the hadrons belonged to the interacting constituents. In comparison this issue is not present in lepton colliders like electron-positron colliders. There the total center-of-mass energy belongs to the two participating fermions.

The distribution of momenta within the protons are described with *parton distribution functions*. These functions are not given as a prediction from the QCD framework but instead need to be determined experimentally from fits to measurements of collision data and they represent the probability density functions that can be used to give the probability of finding a particle carrying a fraction x of the proton momentum at a certain energy scale. These distributions evolve with the proton energy as described by the Dokshitzer-Gribov-Lipatov-Altarelli-Parisi (DGLAP) equations [32–34] so a measurement at one energy scale can be used to make predictions at another scale. The measurements of parton distribution functions are made by compiling data from various experiments at different energy scales as for example is done by the NNPDF collaboration that recently released parton distribution functions for LHC Run II known as the NNPDF3.0 [35].

6.4 Simulating QCD

The highly energetic initial interaction between quarks called *hard interaction* between the quarks can be computed due to the QCD being asymptotically free at high energies. The steps after the initial interaction where *parton showers* form as the quarks or gluons can radiate additional gluons in *splitting* events need to be implemented using approximative methods.

This process can be modelled as a Markov chain and the implementation done using Monte Carlo methods. The implementation details for such methods can vary and in particle physics simulations there are differences in how generators such as Pythia [36, 37] or Herwig [38] treat these simulations. Different showering schemes for QCD interactions are presented in more detail at [39].

Hadronization models determine how the final state colorless hadrons are formed from the gluons and quarks created in the showering step. As hadronization takes place at lower energies where QCD is non-perturbative. They treat the color connected hadrons as a system that will hadronize into its end state collectively. The models are effective in so they contain free parameters that are determined from fits to data. The two major event generators differ in their hadronization models as well. Pythia uses the *string hadronization model* where two color connected quarks are connected by a flux tube that stretches as more energy is

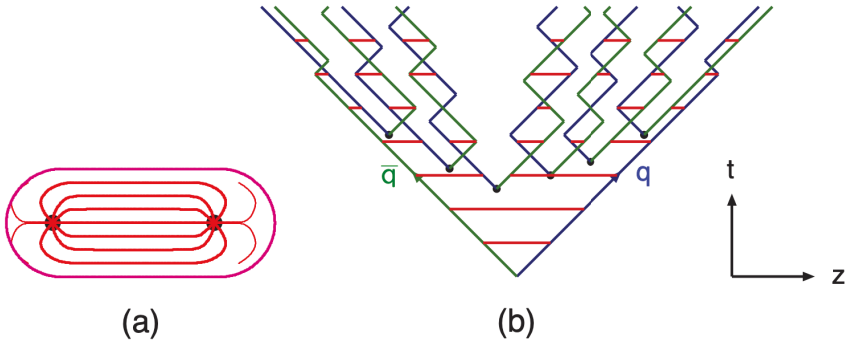


Figure 6.1: **a)** The quark pairs are thought to be connected by a flux tube that stretches as the quarks get pulled apart. **b)** This leads to a model where the hadronization phenomenon is described in terms of relativistic one-dimensional strings (the flux tubes) that get stretched and eventually broken when the energy stored in the string is sufficient to produce a new pair of quarks. Figure from [40]

stored to it as the quarks are pulled apart, until it snaps forming new quarks in the process. The flux tube is represented as a one-dimensional relativistic string that can break into two strings as it is "pulled apart" by the quarks moving further away from each other. The string model concept is visualized in Figure 6.1.

Herwig uses a *cluster hadronization model* relying on preconfinement property in QCD, dictating that the parton shower forms color-singlet combinations called clusters at each stage of the shower. These clusters have an invariant mass distribution that does not depend on the center-of-mass energy of the collision. After the initial color-singlet clusters are formed they will decay into lighter cluster or into hadrons via two-body decays. While the two different effective hadronization models above take a different approach in how to model the phenomenon, both produce similar results and are used in particle physics research.

Due to its non-perturbative nature and self interactions between the force mediating gluons the QCD is arguably the most difficult portion of the Standard Model to simulate. Rigorous derivations from the first principles of the theory often need to be replaced by effective models fitted to data. As such simulating QCD interactions often comes with large uncertainties when used in analyses and often an effort is made to avoid using the direct simulations by using data-driven methods for determining backgrounds from QCD instead. Such method is described in Section 21.1 where a data-driven background measurement is used in the search for the charged Higgs boson.

Chapter 7

Beyond the Standard Model

While the Standard Model is arguably the pinnacle of modern physics combining theoretical and experimental results into a framework that has withheld experimental testing to an incredible degree of accuracy it still has its shortcomings. Some of the issues of the Standard Model can be considered to be more by design than an actual failure of the model like missing gravity as one of the fundamental interactions. Others like the hierarchy problem are more of an unexplained feature of the theory that is of cosmetic nature instead of a fundamental error in the theory. In case of baryon asymmetry the concern is a lack of believable mechanism for providing the extreme asymmetry between matter components in the observable universe. In the following some of the concerns in the Standard Model are discussed as a motivation for further theories extending the SM and after that some possible extensions relevant to the search presented in this thesis are considered.

7.1 Problems of the Standard Model

Hierarchy problem: With hierarchy problem the issue is with the significant degree of fine-tuning required to cancel out divergences in the theory. This has to do with the parameter controlling the Higgs boson mass μ and the vacuum expectation value v. The Higgs potential is sensitive to new physics at any energy scale including the *Planck scale* where effects of gravity become relevant for particle interactions necessitating new physics to be present. However the Higgs boson mass and vacuum expectation value are both orders of magnitude away from the Planck scale $\lambda_{\mathrm{P}} = 10 \times 10^{19}$ GeV requiring careful fine tuning of some parameters of the theory to cancel the effects of these ultra high energy phenomena from affecting the Higgs boson mass and pulling it to higher values. Many theoretical extensions like *supersymmetry* or *extra dimensions* can be used to provide a less suspicious way of cancelling the contributions of Planck scale physics on the Higgs boson mass.

Baryon asymmetry: Also known as the matter-antimatter asymmetry the *baryon asymmetry* is based on the observation that in the observable universe matter seems to be much more common than antimatter. In order to account for this the theory needs to have a process that produces an excessive amount of baryons over anti-baryons, violates the charge conjugation

symmetry and charge conjugation parity symmetry and finally the reaction has to take place out of thermal equilibrium. These conditions are known as the Sakharov conditions [41]. Standard Model provides CP-violation through weak interactions the magnitude of the effect is not considered large enough to explain the disparity of matter and anti-matter [42].

Dark matter: Cosmological observations indicate the existence of a form of matter that is interacting primarily through gravity. This has been seen in multiple independent observations including cosmic microwave background observations, gravitational lensing and galactic rotational velocity curves [42]. The existence of dark matter is generally accepted as a fact regardless of it never having been directly observed in a laboratory environment. The Standard Model include a candidate for this type of non-luminous matter in the form of neutrinos but as with the baryon asymmetry what is already included in the SM is not considered to be enough to account for the observations and some new physics are expected to explain the dark matter phenomena.

Gravity: In order to understand gravity as an interaction between particles a description of gravity in terms of a quantum theory is needed. This is needed for so called Grand Unified Theories (GUT) containing all of physics in a single framework.

These and other issues in the Standard Model are generally considered as a sign that there is something beyond the Standard Model, a theory that will contain the incredibly accurate and predictive Standard Model as a special case but also explain any short comings the original theory might have in terms of some more fundamental theory of physics.

7.2 Supersymmetric theories

When the LHC was being built there was a hope that it would be the machine that discovers the supersymmetry. In supersymmetry new degrees of freedom of the particles on top of the four in regular space-time are added through extending space-time into a *superspace*. These new degrees of freedom are related to the old ones through supersymmetry transformations, leading to a relationship between bosons and fermions: They can be interpreted as two states of particles differing by spin-1/2 [43]. Since there has been no abundance of bosons differing from the observed fermions just by their spins while having the same mass, this supersymmetry is assumed to have been broken and the supersymmetric masses to have been pushed to higher energy regimes not yet observed with colliders.

A Minimal Supersymmetric extension to the Standard Model (MSSM) [44] proposes the hierarchy problem of the Standard Model could be solved using the supersymmetric ideas. The source of the problem are the radiative corrections to the Higgs mass requiring for fine-tuning to cancel out. Now if each particle coupled to the Higgs boson that contributes to the radiative corrections were to have a supersymmetric partner. The boson and fermion contributions to the Higgs mass radiative corrections are to the opposite directions so if both the particles in the supersymmetric pair give equal contribution in magnitude they end up cancelling exactly and remove the need for fine tuning of the Higgs self-coupling term. This assumed supersymmetry would then directly double the particle content of the Standard Model as each already found "regular" particle would require a high mass supersymmetric

partner to be included in the theory.

In addition to allowing a fix for the hierarchy problem supersymmetry gives a new candidate for dark matter. In order to preserve the baryon and lepton number conservation of the Standard Model the supersymmetric reactions would need to preserve so called R-parity where regular particles have R-parity of +1 and all supersymmetric particles -1. This would prevent the lightest supersymmetric particle from decaying at all since it could not produce another supersymmetric particle as a product of the reaction without violating energy conservation laws.

7.3 Extended Higgs sector

In the Standard Model the Higgs sector is said to be minimal because there is only a single Higgs doublet. The reasoning for this is purely practical: with the spontaneous symmetry breaking only one complex scalar doublet was needed to provide the masses for the bosons of the weak interaction and as such explain the short range of the weak interaction through massive bosons while retaining the symmetry of the Lagrangian.

However there is no reason why the Higgs sector could not contain more Higgs doublets. Some Standard Model extensions indeed require the Higgs sector to be extended as well in order to be able to give rest masses to all particles requiring it. The first extension to the Higgs sector and the only one presented in this thesis are the two Higgs doublet models (2HDM) where another complex scalar doublet is introduced, originally presented in [45]. The new doublet adds more parameters that can be tuned for different theories to provide a rich phenomenology in the Higgs sector. This phenomenology allows for example a new dark matter candidate to the Standard Model [46] and a strong first order phase transition that would have an effect on electroweak baryogenesis that could help explain the observed baryon asymmetry [47, 48].

The addition of a new doublet includes new Higgs bosons with masses differing from the Higgs boson that has already been observed to the theory. For the 2HDM models the Higgs sector contains eight degrees of freedom, three of which are used to give masses to the weak interaction mediating $W^{\pm}$ and Z^0 bosons as described before. The remaining five degrees of freedom manifest as five Higgs bosons: Two neutral scalar Higgs bosons one of which is the one that has been discovered, two charged scalar Higgs bosons and one pseudoscalar Higgs boson.

The ratio between the vacuum expectation values of the two Higgs doublets is usually a parameter of interest in theories with an extended Higgs sector, defined as

$$\tan\beta = \frac{v_2}{v_1}. \tag{7.1}$$

Search results for the Higgs bosons in the 2HDM models are often interpreted with respect

to values of this parameter. An observation of another Higgs boson would be a clear signal for physics beyond the Standard Model since there is only one scalar boson included in the Standard Model and it has already been observed. Additionally signals like the ones predicted from the charged Higgs boson can be effectively separated from the background of hadron collisions due to correlations between the helicities of the particles resulting from the decay of the boson and the originating particle. The experimental aspects of the charged Higgs boson searches are discussed in more detail in the chapter dedicated for the search of charged Higgs boson in the fully hadronic $\tau^+\nu_\tau$ decay channel.

Part II

Machine learning

Chapter 8

Basics of machine learning

8.1 Algorithms that learn to solve problems

Machine learning can be considered as the study of algorithms that are able to adjust their behaviour based on data that they are shown. Common features that are shared across the variety of machine learning algorithms in use today are that their aim is to learn how to solve a given problem for instances of data that have not been encountered before, and do this by updating their own parameters based on some rules and a dataset given for training. This makes a machine learning algorithm as much a function of the training data that determine its parameters as it is a function of its algorithmic model.

The general task of a machine learning algorithms falls in the category of pattern recognition i.e. identifying patterns in data that are useful for solving a given task. However the learning portion makes the study of these algorithms a field of its own, as the question of what are the useful patterns is left for the algorithm to find out in a data-driven manner – by looking at the data.

Machine learning tasks can be further classified based on what type of output is required and what type of feedback is given to the algorithm during training. For the context of this thesis the important distinctions are:

Based on output

- **Regression:** The algorithm is required to predict a continuous value based on the given inputs. The task could be predicting what the temperature outside will be tomorrow based on a set of measurements that are available now or the value of an apartment based on it's location, size and other amenities.
- **Classification:** The algorithm needs to classify the input into one of k classes. This could be classifying hand-written digits to categories of integers between zero and nine based on an image or patients into categories of having some disease or not having the disease based on their symptoms and results of medical tests.

Based on training feedback

- **Supervised learning:** The true value that the algorithm is trying to predict is available during the training. This way the algorithm is able to update its state based on some measure of error it makes on its predictions with respect to the true value, often referred to as the **target** or **label**. For example if we have the labels for what the hand-written numbers represent available during the training, the classifier can learn to optimize its predictions on what number is shown in the image to what the provided label says it should be.
- **Unsupervised learning:** There is no targets or labels provided during the training and the algorithm is trying to learn something useful based solely on the input data. An example of an unsupervised learning task could be clustering data into groups of similar data points or ranking data points based on how anomalous they are compared to the other points to perform a type of anomaly detection.

Not all use cases of machine learning fall directly in these classes, especially since many higher level tasks such as driving an autonomous vehicle can combine many sub-tasks like classifying objects or determining at what speed an object is moving, but this gives a general framework to discuss what type of task is being solved. Also any single task is not rigidly bound to one categorization since a regression task of predicting a house's value can easily be turned into a classification task of classifying a house into certain price category.

The problems studied in this thesis will mainly consider classification tasks that are being trained using supervised learning.

8.2 Performance measure

In order to determine if an algorithm fulfills it's task after being trained, there must be a performance measure that quantifies this. For classification it could be the portion of correct predictions called the **accuracy** of the model or one could monitor more comprehensive set of metrics like **recall**, that is the ratio of correct predictions to the class and the total number of samples in the class. It is important to determine a suitable performance measure for a given task early on since it will guide the development of a machine learning solution and finally inform when the algorithm achieves the required level of performance and can be considered ready for deployment.

Usually the goal is to produce an algorithm that performs well on new data that was not used when training the algorithm. For this reason it is common practice to keep a **test set** of data that is independent from the **training set** used in learning the parameters of the algorithm. Performance is usually measured on the test set that does not get to affect the training, and so it simulates how the algorithm would perform on new unseen data.

In many cases the performance metric that one tries to optimize cannot be measured in a practical manner to be used during the training. As an example in this thesis one of the metrics that is optimized is the degree of which the classification shapes the distribution of one of the input variables. This can be quantified when the datasets before and after the classification selection are available, but it is difficult to use as a target while training an

algorithm. In these cases a **surrogate loss function** that is tractable during the training and correlates with the original performance metric can be used.

8.3 Loss functions

8.3.1 Mean squared error and mean absolute error

As a concrete example of two useful **loss functions** that are encountered in many regression problems are the **mean absolute error** (MAE) and the **mean squared error** (MSE).

$$MAE = \frac{1}{N}\sum_{i=1}^{N} |\hat{y}_i - y_i| \,, \tag{8.1}$$

$$MSE = \frac{1}{N}\sum_{i=1}^{N} (\hat{y}_i - y_i)^2 \,, \tag{8.2}$$

where $\hat{y}_i$ is the predicted value and y_i is the target for sample i. While both of these loss functions are minimized by the same values $\hat{y}_i = y_i$ for all samples, and as such they will lead to the same end result when applied to the same dataset if the problem is solved perfectly. In practical problems however the optimal configuration that achieves zero loss is usually never found unless the problem is trivial. Although the end point might be the same, the learning dynamics with the two losses can vary significantly. MSE gives large significance to the samples that are far from the correct prediction while being more lenient with small deviations from the true value when compared to MAE.

8.3.2 Binary crossentropy

For the context of classification, binary crossentropy is a common example of a loss function that will be also used later in the thesis.

$$H(q) = -\frac{1}{N}\sum_{i=1}^{N} y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i). \tag{8.3}$$

Minimizing binary crossentropy is minimizing the negative log-likelihood between the predicted distribution and the label distributions based on the dataset.

8.4 Gradient descent

Gradient descent is an optimization algorithm for iteratively minimizing a function, originally presented by Cauchy in 1848 [49]. It is based on taking small steps towards the local direction of deepest descent which can be found by calculating the negative of the gradient at any point. For convex functions gradient descent is guaranteed to end up near the global minimum. With non-convex functions that optimizing a machine learning algorithm present the gradient descent might end up stuck in a local minimum instead. Algorithm 1 contains a basic form of gradient descent.

Algorithm 1: Gradient descent algorithm.

Result: Returns $x_{\min}$ that minimizes the target function $f(x)$
Let $f(x)$ be a scalar function that is continuous and differentiable for $\forall x$
while *termination criteria not met* **do**
  $\nabla f(x_n) = \left|\frac{\partial f(x)}{\partial x}\right|_{x=x_n}$
  $x_{n+1} = x_n - \lambda \nabla f(x_n)$
  check if $f(x_{n+1})$ satisfies termination criteria;

where λ is the **learning rate** determining the gradient step size. The termination criteria could be either some predetermined value that $f(x)$ needs to reach or it could be related to if the value of $f(x)$ stops decreasing. The algorithm generalizes to multi-dimensional inputs $\mathbf{k} = (k_0, k_1, \ldots, k_i)$ and for non-scalar functions $f(x) = y, y \in \mathbb{R}^j$ in a straightforward manner.

The gradient step size determined by λ is critical on how close to the global minimum the optimization can get in the convex optimization and determining if the algorithm gets stuck in local minima for the non-convex optimization.

An important extensions to the base algorithm especially in the context of machine learning is the Stochastic Gradient Descent (SGD), where an approximation of the true gradient is calculated from a subset of the dataset instead of the whole dataset, leading to significantly reduced memory footprint in computing the gradients. While there is some lack of clarity on who was the first to propose SGD, in the context of machine learning it was first used by Rosenblatt [50]. The same article also presents the **perceptron**, a predecessor of **feedforward neural networks** also called multilayer perceptrons. The idea of stochastic approximations is generally attributed to Robbins and Munro [51]. It is found that using stochastic approximation of the gradient descent algorithm leads to better generalization performance in large-scale learning systems, although it has a slower rate of convergence than the regular gradient descent algorithm [52]. Stochastic gradient descent forms the basis of many of the popular optimization algorithms used in training deep neural networks.

Another useful extension to consider is the inclusion of a velocity term to the update rule in gradient descent as was introduced in [53]. Analogous to the physical momentum, this velocity term is a conservative term in the update rule, pushing the update into the same direction as the last updates have gone and increasing in magnitude if consecutive updates are consistent in direction. That is

$$x_{n+1} = x_n - \lambda \nabla f(x_n) + \alpha \Delta x_{n-1} \tag{8.4}$$

where α is an exponential decay factor between 0 and 1, determining how much the previous gradients affect the current update step. However the same paper reintroduced and popularized another influential idea to machine learning context, the **backpropagation** algorithm.

8.5 Back-propagation

Back-propagation is a method for efficiently computing the gradients needed for updating parameters of a neural network for example. While Rumelhart, Hinton and LeCun [53] have been attributed as having brought back-propagation into the general knowledge in the context of training neural networks, in 2019 The Honda Prize [54] was awarded to Geoffrey Hinton for achievements in creating technologies that enable application of AI including the back-propagation algorithm. This spurred some critique [55] as the idea of back-propagation did exist before. Currently the earliest source attributed with presenting the modern form of back-propagation is the Master's thesis of Seppo Linnainmaa [56].

Regardless of how the claim to fame is shared among the pioneers, the back-propagation algorithm has a central role in modern machine learning as it has fueled a significant portion of the deep learning research of the last decades. While the idea of back-propagation seems like a simple combination of forming computational graphs and using chain rule of derivation to calculate how to update parameters of a function, the saving on computational costs compared to a naive approach are immense. This is demonstrated using Figure 8.1. Let x_1 to x_6 represent intermediate values of a computation. The arrows point the direction of the computation and two arrows pointing to the same node mean summation. Let function f be applied at each edge of the graph, represented by an arrow. For example the value $x_3 = f(x_1) + f(x_2)$. If one is interested in computing the gradient of the output y with respect to any of the nodes, for example x_3, it could be done by applying the chain rule

$$\frac{\partial y}{\partial x_3} = \frac{\partial y}{\partial x_6}\frac{\partial x_6}{\partial x_3} + \frac{\partial y}{\partial x_5}\frac{\partial x_5}{\partial x_3} = f'(x_6)f'(x_3) + f'(x_5)f'(x_3) \tag{8.5}$$

Assuming one stashes the calculation what is the derivative of f, this still requires three unique evaluations of the derivative, two multiplications and one addition so seven operations to calculate $\frac{\partial y}{\partial x_3}$, when the node values x_1 to x_6 are known. Should one choose to calculate the partial derivative with respect to x_1, the same procedure would lead to

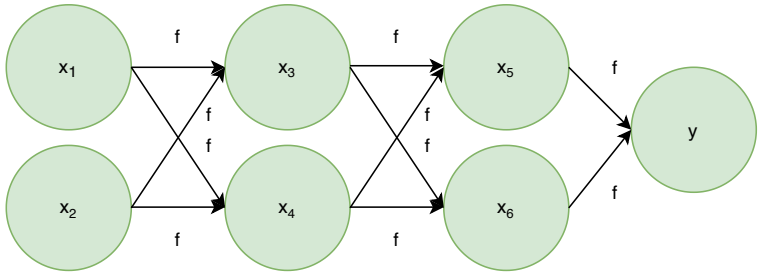


Figure 8.1: Simple computational graph for demonstrating merits of back-propagation.

$$\frac{\partial y}{\partial x_1} = \frac{\partial y}{\partial x_6}\frac{\partial x_6}{\partial x_1} + \frac{\partial y}{\partial x_5}\frac{\partial x_5}{\partial x_1} = \frac{\partial y}{\partial x_6}\frac{\partial x_6}{\partial x_3}\frac{\partial x_3}{\partial x_1} + \frac{\partial y}{\partial x_6}\frac{\partial x_6}{\partial x_4}\frac{\partial x_4}{\partial x_1} + \frac{\partial y}{\partial x_5}\frac{\partial x_5}{\partial x_3}\frac{\partial x_3}{\partial x_1} + \frac{\partial y}{\partial x_5}\frac{\partial x_5}{\partial x_4}\frac{\partial x_4}{\partial x_1} \tag{8.6}$$

$$= f'(x_6)\left(f'(x_3)f'(x_1) + f'(x_4)f'(x_1)\right) + f'(x_5)\left(f'(x_3)f'(x_1) + f'(x_4)f'(x_1)\right) \tag{8.7}$$

$$= f'(x_1)\left(f'(x_6)f'(x_3) + f'(x_5)f'(x_3)\right) + f'(x_1)\left(f'(x_6)f'(x_4) + f'(x_5)f'(x_4)\right) \tag{8.8}$$

$$= f'(x_1)\left(\frac{\partial y}{\partial x_3} + \frac{\partial y}{\partial x_2}\right) \tag{8.9}$$

where the second line shows that one needs to evaluate $f'(x)$ for five values, perform four unique multiplications and three additions leading to total of 12 operations for calculating the gradient. However by grouping the values as shown in the last line, the number of required operations goes down drastically if the gradients with respect to x_3 and x_2 have been calculated earlier and stashed in memory. There would only be one additional evaluation of $f'(x)$, one addition and one multiplication so three operations.

The main merit of back-propagation is utilizing the idea above: If a computation can be organized into a computational graph and the order of computations determined beforehand, the number of operations required to calculate the gradient with respect to the nodes can be significantly reduced if the necessary intermediate values are stashed in memory. If none of the intermediate values are stashed, the number of required operations will scale exponentially with respect to the number of edges in the computational graph due to having to repeat some of the computations over and over again. With back-propagation, the number of operations scales linearly making such computations on large graphs possible. This is seen in formulation of back-propagation algorithm in Algoritm 8.1 where each edge from node $u^{(j)}$ to node $u^{(i)}$ gets visited exactly once to compute the partial derivative $\frac{\partial u^{(i)}}{\partial u^{(j)}}$

Algorithm 2: Back-propagation: Algorithm for computing gradients of output $u^{(n)}$ with respect to the variables $u^{(1)}, \ldots, u^{(n_i)}$ in the graph. All variables are assumed to be scalar. Computational cost of the algorithm is proportional to the number of edges in the graph, assuming computing each partial derivative takes constant time.

Result: Table of gradients GRAD_TABLE of the output node with respect to each node in the computational graph.

Initialize a data structure GRAD_TABLE

$\text{GRAD\_TABLE}[u^{(n)}] = 1$

for $j = n - 1$ *down until* $j = 1$ **do**

$$\text{GRAD\_TABLE}[u^{(j)}] = \sum_{i:j \in \text{Pairs}(u^{(i)})} \text{GRAD\_TABLE}[u^{(i)}] \frac{\partial u^{(i)}}{u^{(j)}}$$

Chapter 9

Deep Neural Networks

Deep neural networks (DNNs) are arguably the most talked about models in machine learning at the moment. They constantly setting new records in perfomance over tasks such as image recognition [57], fooling human perception with generated images of people that do not exist [58] and even drive cars [59]. While the core ideas of neural networks have been around with Rosenblatt's perceptron often hailed as the first neural network in 1958 [50], this latest deep neural network boom required the ideas of Stochastic Gradient Descent and Back-propagation algorithm combined with the substantial increase in computing power for the practical applications to start appearing as well as solving the problem of vanishing gradients with novel activation functions like the Rectified Linear Unit (ReLU) [60]. However already in 1989 LeCun *et al.*[61] built on the newly rediscovered back-propagation algorithm and trained a deep neural network that took images as inputs and recognized hand-written zip codes, demonstrating a real life application with significance in automating mundane tasks with neural networks.

The methods studied for this thesis are also mostly centered around deep neural networks. This is due to neural networks ability to take advantage of high dimensional data and learn useful representations from low-level inputs, the non-linearity of the neural network mappings and their good performance demonstrated in multitude of tasks. At the time when this thesis work began in 2016, the use of deep neural networks in high energy physics was still uncommon although machine learning methods in general were already a staple in the toolbox of particle physicists. During this time deep neural networks have made their way to almost all parts of the reconstruction and analysis pipeline. These applications will be overviewed in Chapter 11. There are important concerns about the black box nature of deep neural network functions in the particle physics community and one of the main subjects of this thesis is to focus on decorrelation methods of deep neural networks that help in producing more predictable neural network behavior that suits the particle physics analysis methods.

This chapter presents the core concepts and terminology of DNNs while the applications will be discussed more in detail in later chapters.

9.1 The need for Deep Neural Networks

One significant advantage deep neural networks and the field of deep learning has over many other machine learning approaches is the ability to take advantage of low-level information and find high-level features without being explicitly programmed to do so. Before deep learning it usually required a domain expert to carefully massage the high-level features out of the data before machine learning could be applied to the problem.

Deep learning also offers the possibility of having **end-to-end** problem solving, where the task doesn't need to be broken up to different stages of pre-processing and converting the output into a usable form. An example would be taking an image directly from a camera as input and producing a text describing what the image depicts as an output.

Deep neural networks are also extremely expressive with the ability to approximate any function arbitrarily well [62]. In more recent works this ability to act as **universal approximators** has been established also in more realistic scenarios with limited width neural networks [63]. Although the ability to express any function using a neural network exists, there is no certain way of learning the parameters required for that. Still it guarantees that unlike in some machine learning models, neural networks do not run the risk of choosing a wrong kernel function or prior distribution that would prevent the algorithm from learning the solution.

The lack of need for pre-processing input data to form domain specific high-level features, combined with the guarantee that neural networks are in principle able to express any function and the possibility to do end-to-end machine learning makes deep neural networks an attractive approach to many problems. Combining these aspects with highly efficient and user-friendly frameworks for training deep neural networks such as PyTorch [64] and TensorFlow [65] explains the recent explosion of research papers and new applications using deep learning.

9.2 Neural network architecture

Deep neural networks are constructed from neurons such as the one depicted in Figure 9.1. Neuron consists of input weights $\mathbf{w} \in \mathbb{R}^n$, a bias term $b \in \mathbb{R}$ and an **activation function** $h : \mathbb{R} \to \mathbb{R}$. Neuron will receive inputs $\mathbf{x} \in \mathbb{R}^n$ either from the previous network layer or from the outside if it is the first layer of the network and compute its output o as

$$o(\mathbf{x}) = h\left(\sum_{i=1}^{n} w_i x_i + b\right), \tag{9.1}$$

where w_i and x_i are the ith components of the vectors $\mathbf{w}$ and $\mathbf{x}$. These neurons are grouped into layers containing k neurons, where k is known as the **width** of a layer. The k neurons are independent of each other, as each one only takes inputs from the previous layers and

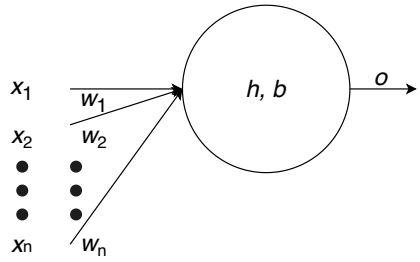


Figure 9.1: Depiction of a single neuron with weights $\mathbf{w}$, bias b and activation function h.

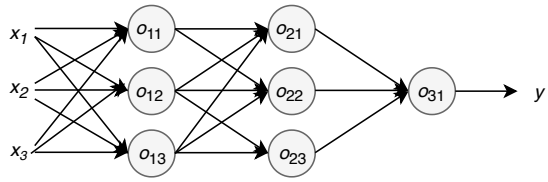


Figure 9.2: A simple neural network consisting of seven neurons, taking input $\mathbf{x} = (x_1, x_2, x_3)$ and producing an output y.

output their results to the following layers. By stacking m of these layers together connecting the outputs from layer $j-1$ as the inputs layer j, one arrives at a neural network of depth m. Width p of the input layer determines the dimensionality of data the network expects and the width q of the last layer determines the dimensionality of the output of the network. The network itself can then be considered as a function $f(\theta, \mathbf{x}) : \mathbb{R}^p \to \mathbb{R}^q$, where θ are the trainable parameters in the neural network, i.e. is the weights and the biases of the neurons. An example of a simple neural network $f(\theta, x) : \mathbb{R}^3 \to \mathbb{R}$ is shown in Figure 9.2. This type of network is also called **fully connected** neural network as all neurons are connected to every neuron of the following layer. In total this network has 21 trainable weights and seven trainable bias terms, so in total 28 **trainable parameters**. Calculating the output y of the network from input $\mathbf{x}$ is called the **forward pass** and calculating the gradients using back-propagation is known as the **backward pass**. When training networks with the SGD algorithms, the training samples are divided into **minibatches** containing usually some dozens of training samples each and the weight updates of the network are calculated based on the whole minibatch instead of individual samples to improve efficiency of the computation.

The activation functions used can be practically any function with a suitable mapping and differentiability so the weight updates of the neural network can be calculated. Often it is desirable that the functions are non-linear, as it allows the network itself to approximate non-linear functions that many of the interesting machine learning problems contain. Other qualities to look out for in activation functions are their saturation and the maximum gradient of the functions as it could lead to problems during training like the **vanishing gradient problem** [66], where subsequental layers containing activations that have their gradient between zero and one get multiplied in the chain-rule quickly reducing the gradient

to zero. Some useful examples of activation functions that will also be used later in this thesis are the **ReLU** (Rectified Linear Unit), **sigmoid** and **Swish** (or Sigmoid Linear Unit) activations

$$ReLU(x) = \begin{cases} x, & \text{if } x > 0 \\ 0, & \text{otherwise} \end{cases} \tag{9.2}$$

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{9.3}$$

$$\text{swish} = x \cdot \sigma(x) \tag{9.4}$$

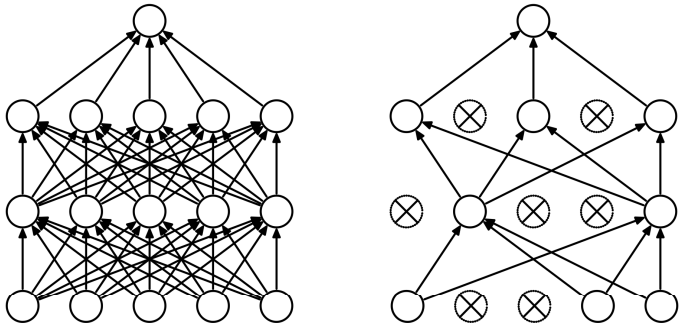
The sigmoid function suffers from the gradient being restricted to values that are less than one, so the above gradient vanishing problem affects deep neural networks with sigmoid activations. However sigmoid is very useful as an activation function for the last layer especially in binary classification problems. The function naturally constrains the output to be between zero and one, simulating predicting probability of the sample label being 1 or not 1. As an added benefit the binary crossentropy often used as a loss in binary classification takes a logarithm of the last activation function, leading to a linear dependence on the input to the last layer which consequently avoids any saturation issues the sigmoid function might otherwise cause when computing the gradient.

The ReLU function is the go-to activation function for the network layers. It has a gradient of one when $x > 0$ and it allows neurons in the network to be shut off, since both the output and the gradient goes to zero if $x \leq 0$. This allows the network to learn configurations where some portions of the network are turned off for some inputs.

9.3 Regularizing the network

The number of trainable parameters tells about the **capacity** of the neural network. As noted earlier deep neural networks are universal approximators and increasing the capacity too much can lead to neural networks fitting the whole training dataset as was shown with an image classification network that was trained on a dataset where the image labels were randomized, removing any real structure the network could learn [67]. This is an extreme example of **over-fitting**, where a machine learning algorithm starts to learn features related to the training data itself like the noise in the inputs, instead of learning a generalized solution to the task. Using a test set to measure the performance after the training exposes over-fitting.

In order to prevent over-fitting from occurring the best approach is to gather more data. If the model has more parameters than there are datapoints in the training dataset there is a good chance of over-fitting occurring. If gathering more data is not an option, the neural network's capacity can be restricted. This is called **regularizing** the machine learning algorithm.



(a) Standard Neural Net (b) After applying dropout.

Figure 9.3: Visualization of the effect of dropout during training. Image from [73].

Most common ways to regularize a network is to add a new term into the loss function that is being optimized during the training. This term increases the loss function for each weight parameter non-zero in the network so that only the weights that really add something valuable to minimizing the actual loss function differ from zero. Two common regularizers are the $\mathbf{l_1}$ or **lasso** (least absolute shrinkage and selection operator) regularizer [68] and $\mathbf{l_2}$ (Tikhonov regularization, ridge regression) regularization [69–72]

$$l_1 = \lambda \sum_i \sum_j |w_{ij}|, \tag{9.5}$$

$$l_2 = \lambda \sum_i \sum_j w_{ij}^2, \tag{9.6}$$

where $\lambda \in \mathbb{R}$ sets the scale of regularization. Both of these terms when added to the function that is being minimized encourage the weights w_{ij} to tend towards zero.

Another often used regularization method is **dropout** [73]. In dropout each neuron and its weights are "shut down" with a probability p during the forward-backward pass cycle. This is depicted in Figure 9.3. In practice this leads to training an ensemble of neural networks with weight sharing but without a significant computational overhead. When producing predictions with the model, all the neurons are used to produce the output but each weight is scaled down by multiplying them by p. This approximates averaging the prediction over the ensemble of networks that was trained.

9.4 Training deep neural networks

As deep neural networks are often non-linear functions their optimization is no longer a convex optimization problem with well defined closed form solutions. Instead iterative approaches

to approximately solve the minimization problem are needed. Training the network in a supervised setting requires four components: A dataset $\mathcal{D}(\mathbf{x}, y)$ containing inputs $\mathbf{x}$ and targets y, a loss function $L(y_{\text{pred}}, y_{\text{true}})$, an optimization method and a model.

The back-propagation algorithm described earlier is the one that enables the training of deep neural networks without a prohibitive computational load. The training dataset is first split into minibatches. Then one by one the minibatches are first propagated through the network in a forward pass and the minibatch loss is calculated. The gradients of this per batch loss are calculated with respect to every trainable parameter in the model. The minibatch gradients can be aggregated and the parameter update is performed after every minibatch has been processed or the parameter updates can be performed after each minibatch at the cost of not being able to process multiple minibatches simultaneously since the model weights need to be kept up to date. This training loop is summarized in Algorithm 3.

Algorithm 3: A simple training loop for a neural network.

Result: Network $f(\theta, \mathbf{x})$ with trained parameters.

Network parameters initialized to random variables θ_0.
Split training dataset $\mathcal{D}(\mathbf{x}, y)$ into minibatches of n samples.
Set number of training epochs $n_{epochs} = k$
for *j=0 up until j=*n_{epochs} **do**
    for *minibatch in minibatches* **do**
        $y_{\text{pred, batch}} = f(\theta_n, \mathbf{x}_{\text{batch}})$
        $\Delta\theta_n = \text{BACKPROP}(L(y_{\text{pred, batch}}, y_{\text{true, batch}}), \theta)$
        $\theta_{n+1} = \theta_n - \lambda\Delta\theta_n$
End training

The training can be terminated in other ways as well, one of the more common methods is **early stopping**. This means stopping the training once some condition is reached. For this reason a **validation dataset** is usually held out during training. The loss function value on the validation dataset is used to decide when to stop training, usually once the loss function value on the validation dataset has stopped decreasing. Otherwise if the training continues there is the risk that the network starts to become over-fitted to the training dataset.

Chapter 10

Decorrelating network outputs

Various applications of neural networks require the network output to be independent of one or more variables. For example a company hiring a new employee or a bank deciding whether to give an applicant a loan might use a deep neural network algorithm to help make the decision. These algorithms could pick up undesirable correlations from the available training data that may lead to illegal discrimination. This is recognized as a significant social and economical issue to be addressed as machine learning algorithms are becoming widely adopted in decision making [74].

Similar needs for decoupling the neural network from one or more variables of interest arise in particle physics as well. It can be used to reduce systematic uncertainties by removing the classifier's dependence on variables with high uncertainties [75, 76], improve discovery significance for new signals by avoiding sculpting background shapes [77] and enhance classifier robustness against varying experimental conditions like the number of simultaneous collisions within a detector [78].

For this work the interest in decorrelating techniques largely lies in the methodology used for discovering new physics processes in collider physics. The method will be discussed in depth in the context of the $H^+ \to \tau^+\nu_\tau$ search at the CMS detector in the chapter 20 of this thesis, but the usual outline of these analyses is as follows: A set of selection criteria is formed with the goal of choosing a subset of particle collision events that is enriched with a significant amount of collisions that contain new physics. The collisions passing this selection are compared to a set of simulated collisions where no new physics are present, that also satisfy the selection criteria. If there is a significant mismatch between the number of real collisions from data and expected collisions based on simulations, this can be considered as a sign of new physics in the data.

However the simulation methods for many of the background processes coming from known physics have large systematic uncertainties attached to them. One way to mitigate the uncertainty is by estimating the background shape in the signal region by interpolating it from a signal-free control region called the side-band. In these side-band regions the good match between simulation and data can be verified and the shape of the distribution in the signal region inferred with small uncertainties. This background prediction can then

be compared to the measured collisions in the signal region. In this chapter it will be demonstrated that a deep neural network classifier will aggressively optimize its performance and can distort the background distribution in the side-band and signal regions in a manner that hinders using this method in predicting the background shapes.

The task of decorrelating the network outputs from variables of interest is discussed, as this can be used to ensure a distortion free distribution that can be used in the background shape estimation. This includes presenting metrics to monitor the amount of correlation, demonstrating how to include constraints in the training process that prevent non-linear correlations between the network decision and a variable of interest from forming and show qualitative results on reducing the shape distortions. A comparison between different methods currently used in the literature is presented.

10.1 A metric to quantify correlation

To prevent a classifier from being correlated to some variable, there first has to be a way for quantifying the amount of correlation between the network output and the variable. As the main goal in decorrelating the classifier in this context is to preserve the shape of the background distribution with respect to a variable of interest, a useful metric would be something that quantifies the similarity between two distributions, before and after the classifier has performed it's selection.

A useful variable to measure such a quantity is found from information theory, in the form of **Kullback-Leibler divergence** [79]:

$$\mathrm{KL}(P||Q) = \int_{-\infty}^{\infty} p(x) \log\left(\frac{p(x)}{q(x)}\right) dx. \tag{10.1}$$

It represents the amount of discriminating information between two distributions P and Q of a continuous variable x and it is also known as the relative entropy between the two distributions. In other words it is a measure of dissimilarity. In this context the form of the equation for discrete probability distributions is more useful

$$\mathrm{KL}(P||Q) = \sum_{x\in\mathcal{D}} P(x) \log\left(\frac{P(x)}{Q(x)}\right), \tag{10.2}$$

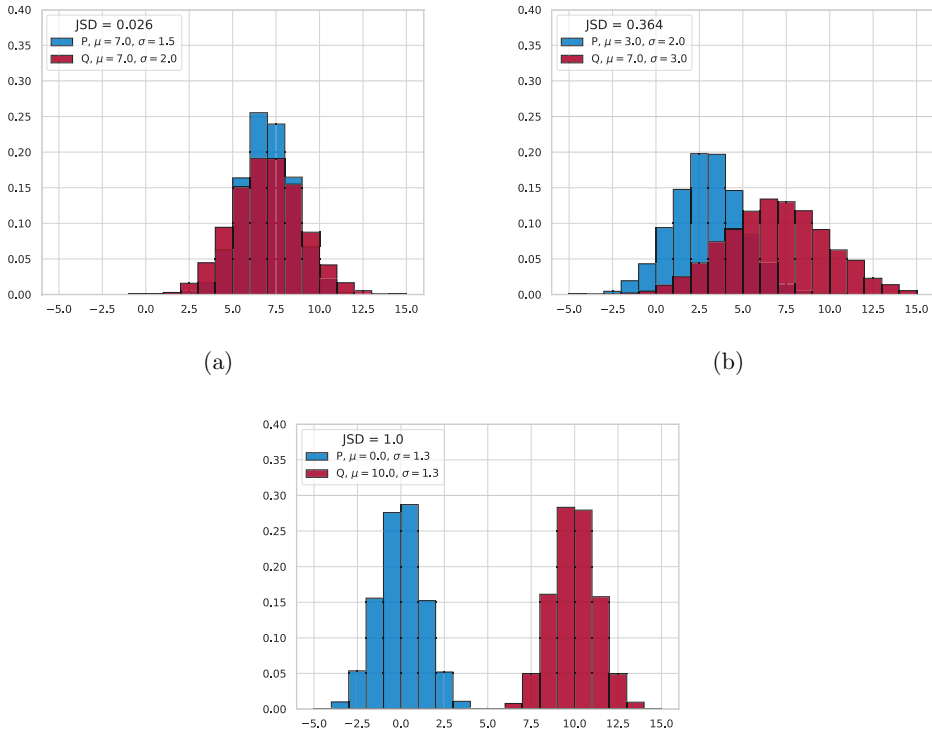
where $P(x_i)$ and $Q(x_i)$ give the likelihood for an observation x_i. This form can be used with normalized, binned histograms and it has the appealing feature of being exactly zero if and only if $P(x_i) = Q(x_i)$ for all $x \in \mathcal{D}$. The Kullbach-Leibler divergence is only defined when $\mathrm{supp}(P) \subseteq \mathrm{supp}(Q)$, which is known also as absolute continuity.

For convenience, it is better still to define a symmetrized and smoothed version of the

Kullbach-Leibler divergence called **Jensen-Shannon divergence**[80] (JSD):

$$\mathrm{JSD}(P||Q) = \frac{1}{2}\mathrm{KL}(P||M) + \frac{1}{2}\mathrm{KL}(Q||M), \tag{10.3}$$

where $M = \frac{1}{2}(P+Q)$. As with the Kullbach-Leibler divergence, JSD will also be zero if and only if the two distributions predict the same value at each data point. If a base 2 logarithm is used in the equation, the upper bound is set to unity. Additionally the requirement for absolute continuity of the Kullback-Leibler divergence can be dropped as JSD is a finite value for any possible value of random variable x in discrete probability distributions P and Q. JSD values between various Gaussian distributions are shown in Figure 10.1 to demonstrate the usefulness of this metric in capturing the similarity between two distributions.



(a) (b)

(c)

Figure 10.1: Jensen-Shannon divergence values on different Gaussian distributions P and Q. The metric captures the similarity between the distributions. **a)** JSD = 0.026 for a very similar pair of distributions, **b)** JSD = 0.364 for somewhat overlapping distributions and **c)** JSD = 1.0 for distributions without any overlap.

10.2 Methods for decorrelating neural networks

With minimizing the JSD as a guide, one can start designing training processes to enforce a classifier to satisfy this restriction while also performing it's original task of classifying as well as possible. Since calculation of the JSD is done at the distribution level, it cannot be directly included in the loss function optimized during training as the learning is done based on gradients and the binning operation does not give a well defined gradient. This issue can be circumvented by approaches such as blurring the binned content with Gaussian filters to ensure differentiable loss function [81] or by computing ensemble level variables using large batch sizes [82]. In the following JSD is used as a metric on whether the selection done by the classifier ends up sculpting the background or not so the lack of gradient is not of importance.

The topic of decorrelating neural networks from specific variables is still a relatively recent topic although it has profound consequences as discussed at the beginning of this chapter. Here is presented three approaches that have yielded a reasonable degree of success in the task: planing [83], adversarial training [77] and distance correlation loss [82]. These methods can be split into two categories where planing is called **data augmentation** and adversarial training and distance correlation loss are **training augmentation**.

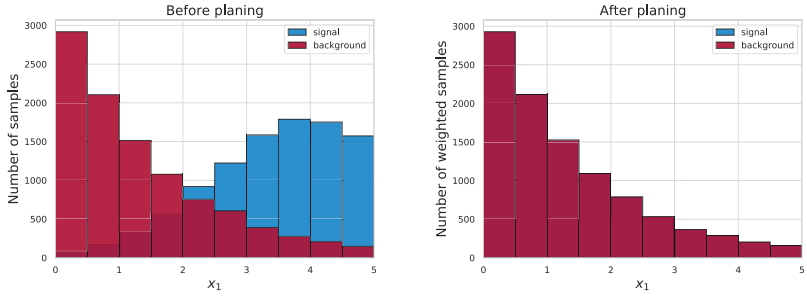
Planing: The input data used for training is weighted so that the distributions for the variable of interest between the different classes are identical. In practice this means that a weight $w_{i,C}$ for event i in of class C can be determined by forming a histogram of variable of interest x, where n_j is the number of events in bin j, so that

$$w_{i,C} = \frac{A_C}{n_j} \tag{10.4}$$

where A_C is the per class normalization factor. As this is done on binned data, the method is only approximate for finite bin width and additionally requires that $n_j \neq 0$ for all bins j in order for $w_{i,C}$ to be well defined.

These weights are then used when calculating the per sample contribution to the loss during training, so the network "sees" a similar effective distribution of x between the classes. Planing is demonstrated in Figure 10.2. Even though planing does little in the way of explicitly decorrelating the output, it is found effective in some problems [84] delivering performance similar to more complicated methods like the adversarial training.

Adversarial training: In adversarial training one uses another neural network to try and infer the value of the variable of interest from the output given by the classifier. The loss function describing the performance of this adversarial neural network is then included in the training to create a combined loss function for optimizing both neural networks at the same time, classifier to predict the class of the sample and the adversary to predict the variable of interest from the output of the classifier. By reverting the back-propagated gradient



(a) (b)

Figure 10.2: Demonstration of planing signal and background with respect to variable x_1, so that the signal events are reweighted to match the background distribution. However as will be shown in the context of a toy model, this does not remove non-linear correlations with x_1 from the other variables.

optimizing the adversary using a gradient reversal layer [85] before passing it through the classifier, the classifier is guided towards a configuration where it tries to produce predictions that the adversary cannot use for predicting the variable of interest.

With the connected classifier and adversarial networks, the training objective becomes that of finding the parameters

$$\theta_C^* \theta_A^* = \arg\min_{\theta_C} \max_{\theta_A} L_C(\theta_C) - \lambda L_A(\theta_A, \theta_C) \tag{10.5}$$

that is, the goal is find the parameters that minimize the loss of the classifier L_C while maximizing the loss of the adversary L_A. Challenge with the adversarial training is the same that is faced with the very popular class of neural networks Generative Adversarial Neural networks (GAN) [86] as well: as both networks are optimized at the same time, the problem becomes unstable and this introduces difficulties in converging to a configuration with good performance in a consistent manner. Countering these difficulties in the training is an active area of research [87–89] and a lot of hand-tuning of hyperparameters is usually to be expected with this type of training.

Distance correlation: Usage of distance correlation metric for the decorrelation problem is a very recent advance in the field [82]. The issue with the Jensen-Shannon divergence is that it requires either binning of the samples or information about the underlying distribution that cannot be really calculated from a single sample. This prevents the usual gradient back-propagation from being used to train the network.

Distance correlation metric was developed in [90–93] to provide a test of independence be-

tween two random vectors that is easy to implement in arbitrary dimensions:

$$\mathrm{dCorr}^2(X,Y) = \frac{\mathrm{dCov}^2(X,Y)}{\mathrm{dCov}(X,X)\mathrm{dCov}(Y,Y)}, \tag{10.6}$$

where $X \in \mathbb{R}^p, Y \in \mathbb{R}^q$ are random variables. The **distance covariance** is defined as

$$\mathrm{dCov}^2(X,Y) = \int_{\mathbb{R}^p}\int_{\mathbb{R}^q} |f_{X,Y}(s,t) - f_X(s)f_Y(t)|^2 \, w(s,t)\mathrm{d}s\mathrm{d}t \tag{10.7}$$

with $f(x), f(y)$ being the characteristic functions of X and Y respectively and $f(x,y)$ is the joint characteristic function for X and Y. The weight function

$$w(s,t) \propto |s|^{-(p+1)} \, |t|^{-(q+1)} \tag{10.8}$$

can be defined up to normalization by requiring dCov to be invariant under constant shifts and orthogonal transformations, and equivariant with respect to scale transformations [94]. Distance covariance can be rewritten in a more practical form as

$$\mathrm{dCov}^2(X,Y) = \langle |X-X'| \, |Y-Y'| \rangle + \langle |X-X'| \rangle \langle |Y-Y'| \rangle - 2 \langle |X-X''| \, |Y-Y''| \rangle \tag{10.9}$$

with $|\cdot|$ referring to the Euclidean vector norm, $<\cdot>$ to the expected value and (X,Y), (X',Y') and (X'',Y'') being i.i.d. samples drawn from the joint distribution (X,Y). This formulation allows for computing dCov^2 and therefore dCorr^2 from a dataset of samples $(\vec{x}_i, \vec{y}_i)$ in a manner that can be used in training neural networks.

Distance correlation has the attractive quality of being bounded between 0 and 1, with $\mathrm{dCorr}^2(X,Y) = 0$ i.f.f. X and Y are independent. It can be included in the network training by modifying the loss function L of the classifier to include a term proportional to the distance correlation between the prediction and some variable to decorrelate from

$$L = L_{\text{classifier}}(\vec{y}_{pred}, \vec{y}_{true}) + \lambda \cdot \mathrm{dCorr}^2(\vec{a}, \vec{y}) \tag{10.10}$$

where $\lambda \leq 0, \lambda \in \mathbb{R}$ is the hyperparameter controlling how the importance of correct classification against keeping the predictions decorrelated, $\vec{y}$ is the classifier output over a minibatch of samples, $\vec{y}_{\text{true}}$ are the true class labels and $\vec{a}$ are the variables of interest of the samples.

Compared to methods like the adversarial training, distance correlation is significantly simpler with only one new hyperparameter to optimize instead of a whole new network to optimize.

10.3 Toy model example

In the following section the three introduced methods for decorrelating network output from a variable of interest is studied in a toy model example with only a few variables with simple relationships between each other. This is done in order to demonstrate the shape distortion of the background distribution caused by a naively trained classifier when applied as a selection, establish planing approach as an easy-to-implement benchmark and the effectiveness of both the adversarial training and the distance correlation approaches. Additionally the low dimensional space allows a informative visualization of how the samples are distributed.

Let each sample in the toy model have four real valued features $\mathbf{x} = (x_1, x_2, x_3, x_4), \mathbf{x} \in \mathbb{R}^4$ and a binary class label $y = \{0, 1\}$. The random samples $(\mathbf{x}, y)$ are drawn as follows

$y = 0$:	$y = 1$:
$x_1 = \text{Trunc}(\text{Exp}(\lambda = 1.5), 0.0, 5.0)$	$x_1 = \text{Trunc}(\mathcal{N}(\mu = 4.0, \sigma = 1.5), 0.0, 5.0)$
$x_2 = \cos x_1$	$x_2 = \cos x_1$
$x_3 = 2x_1$	$x_3 = 2x_1$
$x_4 = \mathcal{N}(\mu = 0.0, \sigma = 1.0)$	$x_4 = \mathcal{N}(\mu = 1.0, \sigma = 1.5)$

where $\mathcal{N}(\mu, \sigma)$ is the normal distribution, $\text{Exp}(\lambda = 1.5)$ is the exponential distribution with $\lambda = 1.5$ and Trunc(a, min, max) signifies that the drawn values are truncated between [min, max] to match the range of the x_1 for $y = 0$ and $y = 1$ cases. When generating the samples, if a value x_1 outside the truncation interval, the sample is rejected and a new draw is made. The resulting distributions for both classes are shown in Figure 10.3.

Let x_1 be the variable of interest. It can be seen from the given definitions that if the classifier predictions are to be completely decorrelated from the variable of interest, it needs to be optimized so that it will not use x_1, x_2 or x_3. Out of these x_2 represents a non-linear dependency and x_3 a linear dependency on the variable of interest. x_4 is the variable that contains the only uncorrelated information useful for discriminating between the samples drawn from the two classes.

In all of the following cases, the neural network architecture for the classifier will be very simple with a limited number of trainable parameters, as the classification task is not very complicated. The network has two hidden layers with 16 neurons each using ReLU activations and a sigmoid output layer predicting the class, $\hat{y}$. The used optimizer is Adam with learning rate lr= $3 \cdot 10^{-4}$. The optimized loss is the binary crossentropy between the true and predicted labels.

Naive classifier: As the name suggests, a naive classifier is optimized directly to solve the classification task using the input variables $\mathbf{x} = (x_1, x_2, x_3, x_4)$ to predict the label $y = \{0, 1\}$. Results of the classifier are shown in Figure 10.4.

The naive network learns to classify the events and achieves a AUC ROC value of 0.951 on an

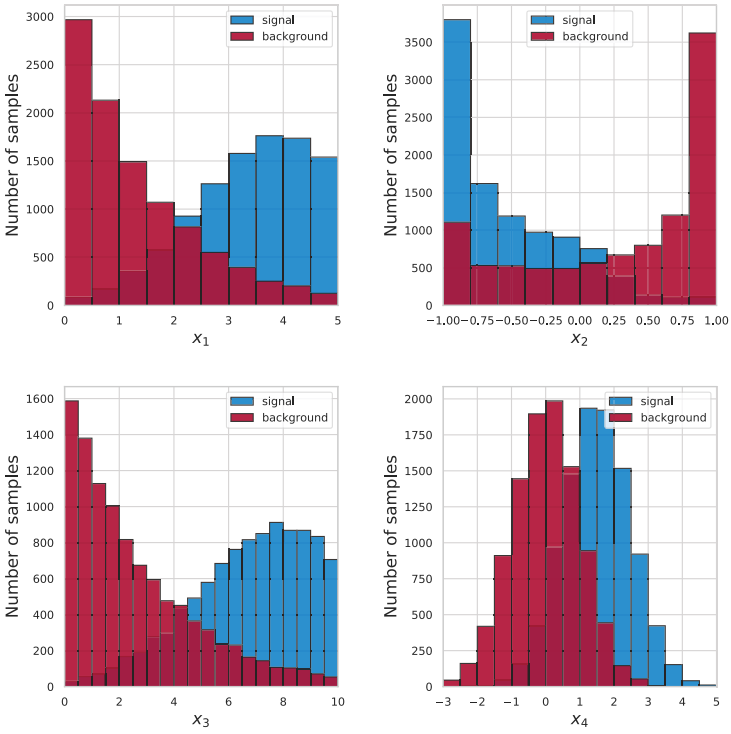


Figure 10.3: Toy model feature distributions for background ($y = 0$) and signal ($y = 1$), both with 10000 samples. x_1 is chosen as the variable of interest and the classifier output is decorrelated from it.

independent test set, the AUC curve is shown in top right plot. Comparing the distribution of output values for signal and background with respect to the variable of interest on top left, the trend with respect to distribution of x_1 in Figure 10.3 is clear. The predictions towards the higher x_1 are dominated by the signal there affecting the predicted values of background as well.

When performing classification a working point is chosen, often using the ROC curve to guide the selection. Every output above the threshold is selected as signal-like and everything below is discarded. For illustration, let the threshold be chosen as $\hat{y} = 0.5$ and bottom left plot demonstrates the effects of this classification on the shapes of the signal and background distributions with respect to the variable of interest x_1. The JSD value which was presented as a measure of how much the selection shapes the distribution is determined to be JSD$= 0.397$ between the background distribution before and after the cut on the bottom right plot. As the after selection distribution visually demonstrates, using the side-band region in x_1 distribution to extrapolate the background shape to the signal region cannot be done in any

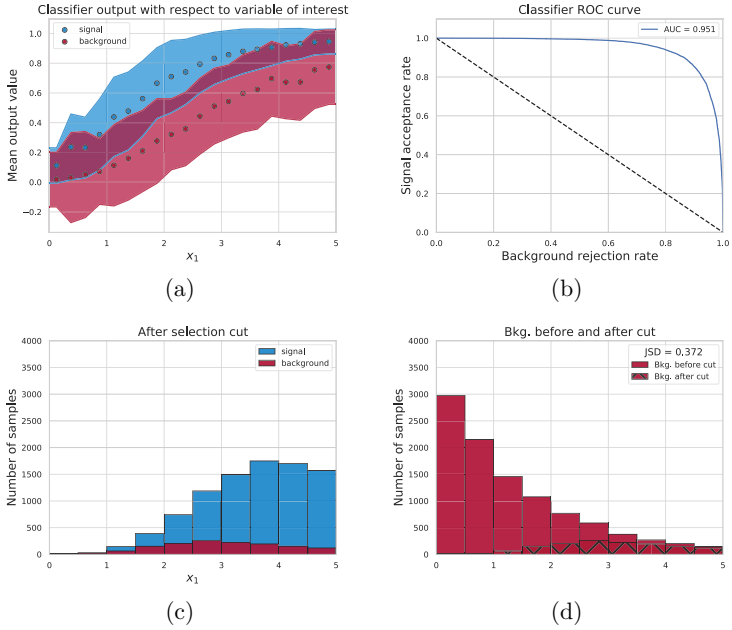


Figure 10.4: Performance plots for the naive classifier. **a)** Mean classifier output and standard deviations from mean binned with respect to variable of interest. **b)** ROC curve for the classifier trained with planing weights. **c)** Distributions of signal and background after the selection cut at $\hat{y} = 0.5$. **d)** Background distribution before and after the selection, where the side-band region is completely cut out.

meaningful way.

Planing: The shapes of the weighted event distributions using the planing weights are shown in Figure 10.5.

One has to be careful with the choice of binning used in with the variable being planed. The used bin width in the planing variable should be small to avoid binning effects from quantizing continuous variables but large to prevent limited per bin statistics from causing significant fluctuations. Both effects are demonstrated in Figure 10.6, where the different bin widths and the number of samples are varied. It should be noted that this problem can be avoided by using more sophisticated techniques to perform unbinned weighting on the events as shown in [95].

Information removed from the input distributions reduces the achieved classification performance, but leads to decorrelation of the classifier output and the planing variable. Results of training the classifier network while using the planing weights are demonstrated in the plots of Figure 10.7.

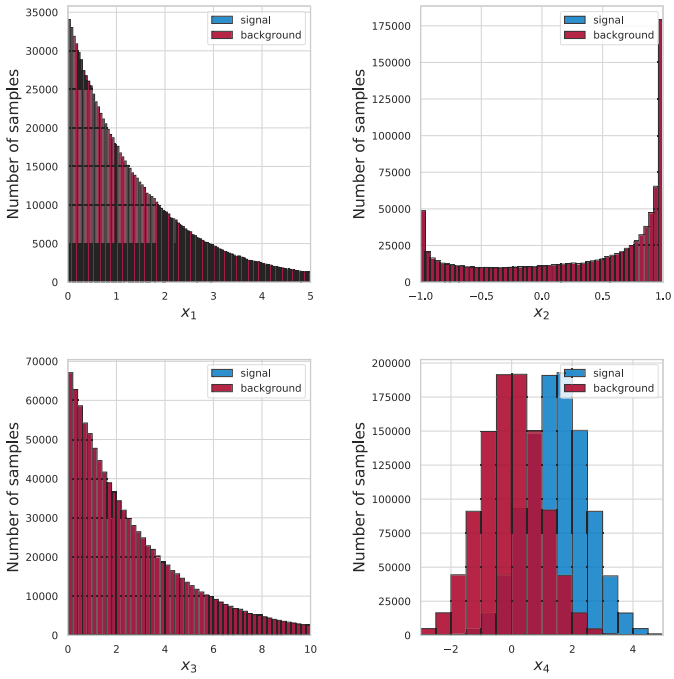
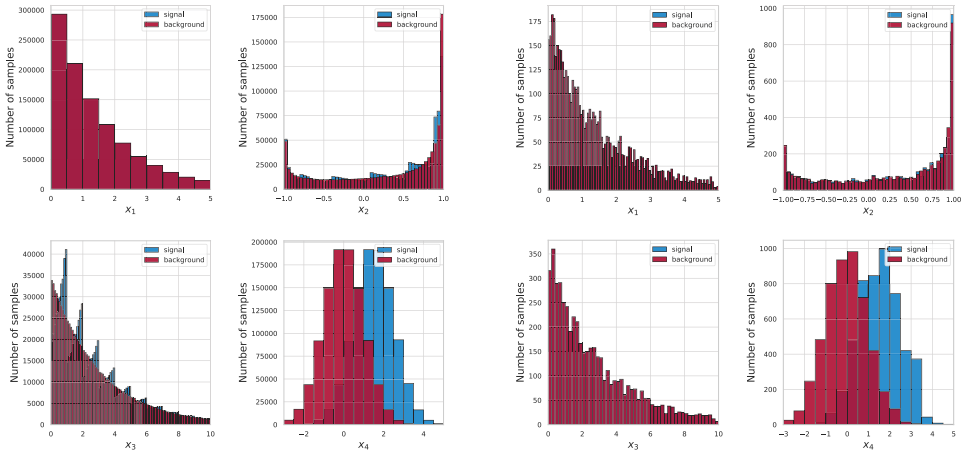


Figure 10.5: Using planing removes the x_1 dependence between signal and background in the input distributions.

Success of the decorrelation procedure in the training is clearly visible, leading to JSD value of 0.001 between the background distributions before and after the selection cut at $\hat{y} = 0.5$. As the shape of the background is preserved, also the side-band region that can be used for estimating the background in the signal region is still present. However the classification performance decreases to ROC AUC = 0.867.

Adversarial training: Using a very simple adversarial with two hidden layers with 16 and 20 components respectively, which is used to fit a Gaussian Mixture Model (GMM) with 10 components. predicting the value of x_1 using the classifier output. The gradients from the adversarial network are reversed when passed to the classifier, leading the classifier away from configurations that help the adversarial in making correct predictions. The adversary is trained in tandem with the classifier, minimizing the negative log-likelihood between the classifier output and the input x_1.

Training adversarial networks is difficult mainly due to the two networks optimizing different tasks at the same time so there are little guarantees on the joint optimization problem



(a) (b)

Figure 10.6: **Left:** Too wide bins used in the planed variable, leading relics in the x_2 and x_3 variables. **Right:** Too small per bin statistic in the planed variable, leading to relics visible in x_2 distribution.

converging into a good minimum. This results in a lot of tuning of the hyperparameters by hand. Additionally the adversarial network introduces a large number of new hyperparameters to tune. Common approaches include using pretraining for the classifier and the adversary before the joint optimization, training both networks at different rates or using different batch sizes for both models.

However the advantage of using adversarial training over planing comes from being able to adjust the training parameters to different levels of decorrelation. Also planing removes the information from the input variables themselves, where as adversarial training pushes the classifier to find a configuration that produces decorrelated predictions while being able to use all of the information in the input data.

Performance of the adversarially trained classifier is summarized in Figure 10.8. There we can see that while the decorrelation having JSD value of 0.002 is not as total as in the planing case, it produces a better classification performance with ROC AUC = 0.887. This still leaves a usable side-band region in the distribution of x_1, as the amount of distortion in the background is near minimal. While in this case the gain from a significantly more work intensive training procedure compared to planing seems a bit underwhelming, in a more realistic scenario with more complex relationships between the used inputs the advantage of adversarial training solution can be more substantial.

Distance correlation: With distance correlation, only the loss function is augmented by adding a loss term that drives the training towards configurations where the distance correlation metric goes to zero. This also allows for a more flexible training where the performance-

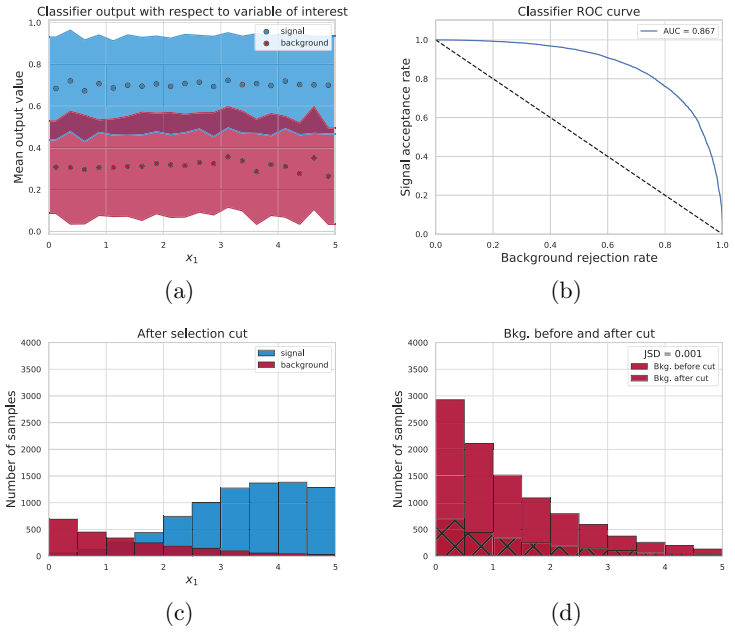


Figure 10.7: Performance plots for the planed classifier. **Top left:** Mean classifier output and standard deviations from mean binned with respect to variable of interest. **Top right:** ROC curve for the classifier trained with planing weights. **Bottom left:** Distributions of signal and background after the selection cut at $\hat{y} = 0.5$. **Bottom left:** Background distribution before and after the selection, where the shape is preserved.

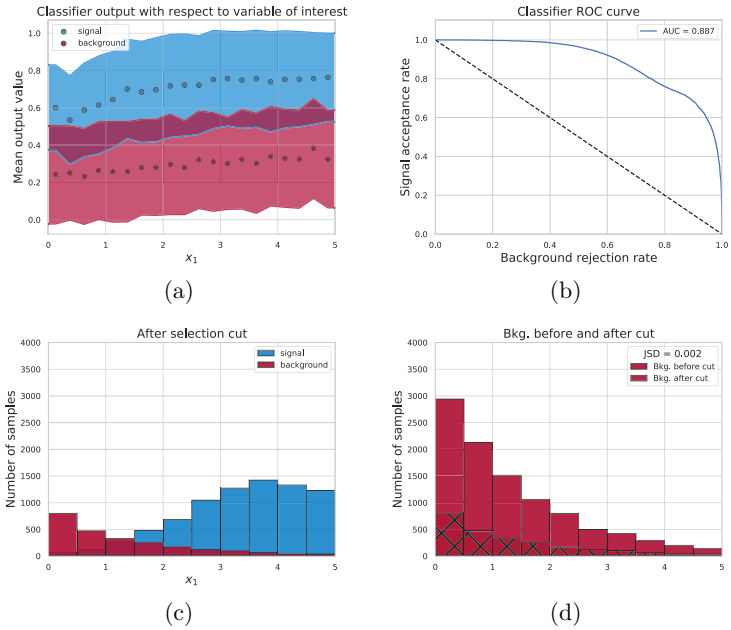


Figure 10.8: Performance plots for the adversarial classifier. **Top left:** Mean classifier output and standard deviations from mean binned with respect to variable of interest. **Top right:** ROC curve for the classifier trained with planing weights. **Bottom left:** Distributions of signal and background after the selection cut at $\hat{y} = 0.5$. **Bottom left:** Background distribution before and after the selection, where the shape is preserved.

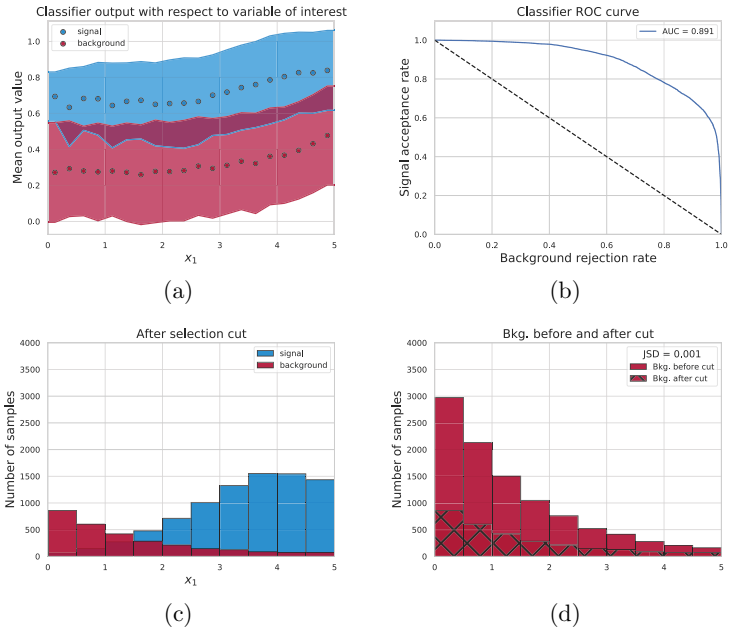


Figure 10.9: Performance plots for the distance correlation classifier. **Top left:** Mean classifier output and standard deviations from mean binned with respect to variable of interest. **Top right:** ROC curve and ROC AUC for the classifier. **Bottom left:** Distributions of signal and background after the selection cut at $\hat{y} = 0.5$. **Bottom left:** Background distribution before and after the selection, where the shape is preserved.

decorrelation trade-off can be controlled with a single hyperparameter that determines the scale of the distance correlation term in the loss function. Also like with the adversarial training there is no information removed from the input distributions, allowing for better classification performance.

In comparison to training the adversarial network, the training process is as simple as with the naive classifier. The tunable hyperparameter controlling the decorrelation has a wide range of acceptable values that work in a predictable manner either increasing or decreasing the degree of decorrelation achieved during the training. The performance of the classifier is shown in Figure 10.9. The classifier maintains JSD value of 0.001, the same as the planing classifier. However the classification performance measured with the ROC AUC is 0.891, exceeding both the planing and the adversarial training examples. The side-band region and the background shape are preserved in the x_1 distribution.

To summarize, three different decorrelation methods were studied in the context of a toy model. While the naive classifier has clearly the best ROC AUC value, selection cut with it shapes the background distribution in a manner that prevents the usage of the side-band region to estimate the background shape in the signal region.

Planing, adversarial training and distance correlation based loss are all able to prevent this background shaping at the cost of the classifier performance. Adversarial training and distance correlation loss allow for more flexibility in the performance-decorrelation tradeoff than planing. However adversarial training in practice is notoriously difficult and training the models is significantly slower. Anecdotally, optimizing the adversarial training for this simple toy model took half a day in order to get the hyperparameters right while optimizing the distance correlation loss training took ten minutes. When approaching more realistic machine learning problems, it is unlikely that the adversarial training would become any easier.

Due to these considerations the real physics analysis use case presented in this thesis will focus on a classifier trained with a distance correlation enhanced loss.

Chapter 11

Machine learning tools at the LHC

High energy physics has been an early adopter of many machine learning methods. The field combines the availability of large amounts of high dimensional data with highly sophisticated simulations to model it, enabling the use of various machine learning approaches to solve interesting problems.

On the other hand the computational demands of reconstructing the collected data in particle collisions require the development of algorithms that are able to maintain good performance but also scale well to increasing number of simultaneous collisions in the detector. The current algorithms have excellent physics performance, but the required computing resources scale poorly with the increasing luminosity in the LHC, especially towards the HL-LHC [96].

Development and deployment of machine learning tools across the pipeline of operations required for a modern particle physics is recognized as one solution to these challenges [97]. This includes topics from data quality monitoring to object reconstruction and offline data-analysis. In this chapter the various machine learning activities taking place around the LHC and its different experiments are discussed. There are also applications in the theory side of particle physics where for example already in 2002 neural networks were used to find unbiased parametrization of deep inelastic structure functions [98] and Neural Network Parton Distribution Functions (NNPDFs) where the PDF can be determined from global fit to data without assumptions on the shapes of the distributions [99]. However here the focus is on the usage of machine learning in the experimental side of high energy physics.

11.1 Trigger-level reconstruction

Performing decisions in the trigger-level requires algorithms that are both fast and relatively accurate as the negative decision means data will be lost. Due to the requirement for fast decisions, Field Programmable Gated Arrays (FPGAs) are used for trigger level computations. A recent trend has been to run machine learning algorithms on FPGAs for the trigger decisions, allowing the high-dimensional multivariate analysis to be run in time scales of hundreds of nanoseconds for the L1 trigger or tens of milliseconds for the HLT trigger.

The concept of deploying neural networks to trigger level using FPGAs is presented in [100] using a case study of a jet classifier that is able to perform inference in $\mathcal{O}(100)$ ns. Additionally the HSL4ML compiler package that makes building neural networks for FPGAs more accessible is introduced by the authors. While in the publication the HSL4ML only supports fully connected neural networks, it represents already an important proof of concept on being able to run a real neural network in the time scale required for L1 trigger decisions.

The CMS experiment uses a Boosted Decision Tree in reconstructing muon energies in the L1 muon endcap trigger [101]. This Endcap Muon Track Finder algorithm also runs on FPGAs, using a BDT that has been trained offline and converted into a look-up table with 2^{30} different patterns stored. As high p_{T} muons are present in the LHC processes involving the weak interaction, being able to confidently identify these events at L1 trigger level is important. This was the first machine learning algorithm implementation in the L1 trigger level at the LHC and it provided a factor of three reduction in background events with muons under the threshold p_{T} being accepted.

Another point of view to improving the triggers with machine learning is presented in [102]. A Variational Autoencoder network can be used to identify anomalous events based on the reconstruction error from passing through the network. As different physical processes introduce different correlations between measured variables, an autoencoder trained with SM physics should reconstruct measurements from BSM signal poorly. This has the advantage of not being tied to any particular BSM process, but instead it works as a filter to catch any non-SM like event. By tuning the cutoff value on the reconstruction error, such a network can be set to select some small number of events daily that it considers the most interesting. If deployed to the the HLT trigger level, the network would be able to select a signal-like subset of events from significantly higher amount of data than any usual offline analysis would.

11.2 Data quality monitoring

Data collected by the experiments is monitored for quality. This is done both as online monitoring where the experts are able to adjust the detector during data taking in order to fix any issues with the measurements and offline monitoring where collected data gets flagged as good quality to be used for physics analysis or poor quality to be left out. These tasks require a considerable amount of person power who in many cases might be performing fairly simple tasks of looking at histograms of summary statistics to try and spot anomalies compared to some baseline. This sort of anomaly detection is a popular type of machine learning problem.

Automatizing this task represents two-fold benefits of releasing the persons responsible to work on more meaningful tasks and being able use more high dimensional inputs to estimate the quality of the data and spot possible detector failure modes faster. A study on automated data quality system for the CMS experiment is presented in [103]. Here the authors use a Gradient Boosted Decision Tree to classify the 2010 data collected by the experiment. The classifier can be trained based on the expert decisions on which data is good and which is bad since all of the data has already been classified by hand. The algorithm takes a fixed amount

of features from every subset of data to be classified such as the kinematic variables of the objects of interest and the coordinates of the origin. During training it learns to minimize the fraction of datasets that are rejected and passed on to a human expert for evaluation while under constraints to keep false negative and false positive rates under some predetermined thresholds. The authors that they are able to reliably process at least 20% of the samples, and effectively reducing the need for a human expert to intervene by the same amount.

With a similar goal deep neural networks and convolutional autoencoders are trained to catch both known and unexpected anomalies respectively in the muon DT subdetector of the CMS experiment [104]. Both approaches are based on detecting anomalous patterns in the measured energy deposits, which can signify failure modes in the detector elements. An example of an occupancy map of one DT chamber is shown in Figure 11.1. It displays the measured particle counts in different parts of the chamber when operated at different voltages in A and B. This is a typical misbehavior that the algorithms used before could not automatically detect, but the machine learning approach is able to capture as it is able to compare the occupacy patterns between the layers within a chamber from the image.

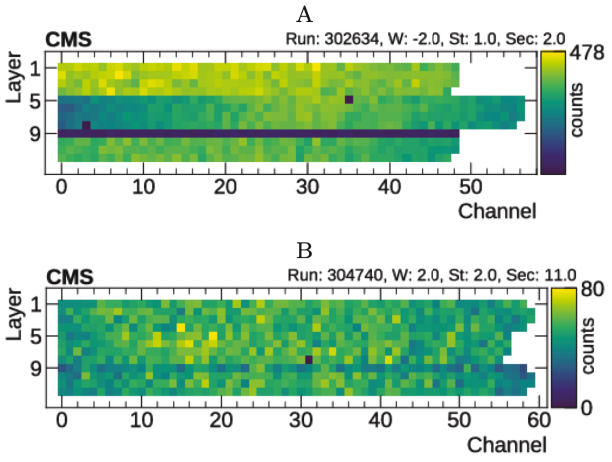


Figure 11.1: Occupation maps of one muon DT chamber at voltages 3200V(A) and 3450V(B). Layer 9 can be seen misbehaving in both in a manner that would not have been caught by the regular data quality monitoring algorithms. Figure from [104].

The data quality monitoring algorithms for the muon DT were comissioned with the early data of 2018. Although the methods were developed for the muon DT, the authors note that similar approach should be applicable for other subdetectors as well.

11.3 Fast simulation

To simulate the proton-proton collisions with pile-up collisions requires significant computational resources. Especially the simulating the interactions of the particles with the detector materials consume a large fraction of the computational budget available.

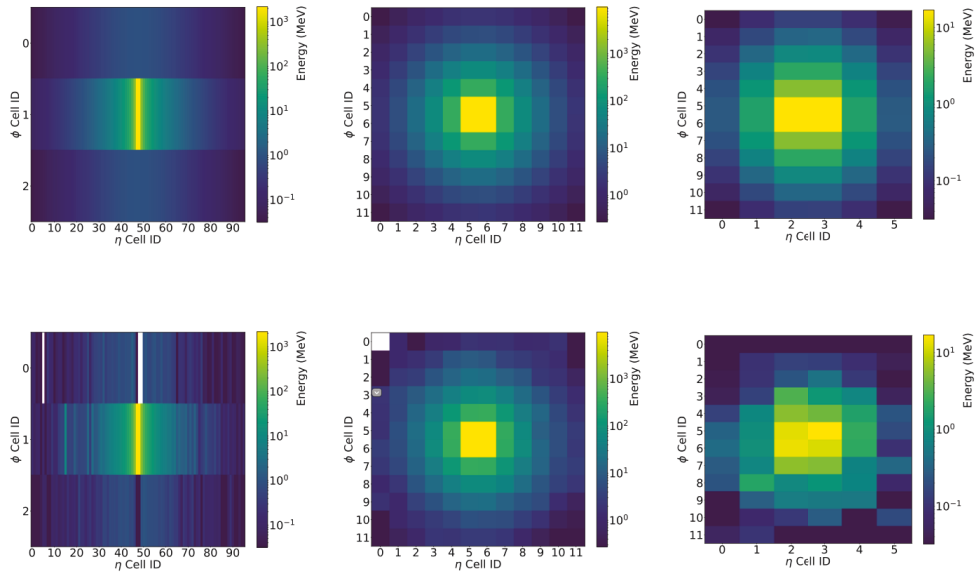


Figure 11.2: Average γ shower energy distributions, shown for increasing calorimeter depth from left to right. **Top row:** GEANT4 **Bottom row:** CALOGAN. Figure from [108].

Conventional way to produce faster simulations for purposes where the full physically motivated simulation is not necessary is by parametrizing the detector effects and utilizing the Monte Carlo truth to speed-up the simulation process by $\mathcal{O}(100)$ [105]. Deep learning and Generative Adversial Networks (GANs) offer an alternative approach where a generative model G is trained to map an input of random noise to a sample produced by a simulation. GANs have shown tremendous success in tasks like generating photograph like images of faces from noise [106].

Speeding up the task of modeling the interactions between particle jets and the calorimeters measuring is studied in [107] as a task to produce "jet images" that contain the distribution of energy in a 2D image as if it were a single layer calorimeter. While being able to reconstruct some of the jet variable distributions reasonably from these images, an important issue is raised where the GAN is not able to produce W jets that end up looking like QCD jets and vice versa. This is suspected to be caused by the loss formulation of the adversarial training favouring unambiguous images.

The concept is extended in [108] to generate more realistic 3D particle showers in multi-layer calorimeters instead of just one 2D image. It was demonstrated for γ, π^+ and e^+ showers and again a reasonable matches in various distributions are seen, however the matching is not quite agreeable everywhere. Examples of average γ shower energy distributions at different depths of the calorimeter are demonstrated in Figure 11.2 where one can observe qualitative similarities and discrepancies between GEANT4 and the CALOGAN produced showers.

The method is considered to show great potential despite its limitations as it offers a speed-up factors of up to $\mathcal{O}(10^5)$ compared to the standard simulation methods. The technique is still under development with further studies in QCD dijet production in [109] showing promising steps towards improving the faithful reproduction of the jet variable distributions and a methodology for simulating any generic calorimeter at [110] that provides a recipe for producing GANs to simulate calorimeters.

11.4 Tracking

Reconstructing the flight paths of charged particles is recognized as one of the dominant contributions in the required computing budget for event reconstruction at the CMS detector [111]. This is shown in Figure 11.3. Crucially many of the algorithms used in seeding and track building with excellent physics performance will scale quadratically or worse with respect to the number of hits in the tracker [112]. This is due to the inherently sequential nature of the algorithms being used. The tracking algorithms are presented in more detail in Section 16.

The issue is known and there are significant efforts put in both recasting the current algorithms to a parallelized and vectorized form that can take advantage of modern CPU and GPU resources [113–118] and studying machine learning approaches that would lead to better scaling in the HEP.TrkX [112, 119] and EXA.TrkX projects [120].

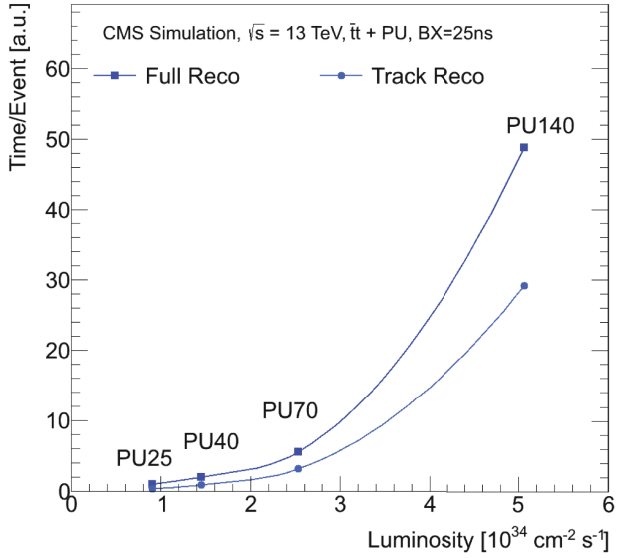


Figure 11.3: Time required for the full reconstruction and just the track reconstruction of a collision event in the CMS. Figure from [113].

Hit clustering, seed selection, track building and track fitting are all viable candidates for

new machine learning based solutions. HEP.TrkX explored the track building problem with various approaches. They noted the tracking subdetector can be presented as an input sequence of pixel arrays where each layer of the detector is a step in the sequence. This sequence can then be used to train a Long-Short Term Memory (LSTM) neural network. This is demonstrated in Figure 11.4. The LSTM networks are similar to Kalman Filters in the sense that each input in the sequence updates the state of the estimator. The model can predict the following hit belonging to the track based on the information it has captured from the earlier layers.

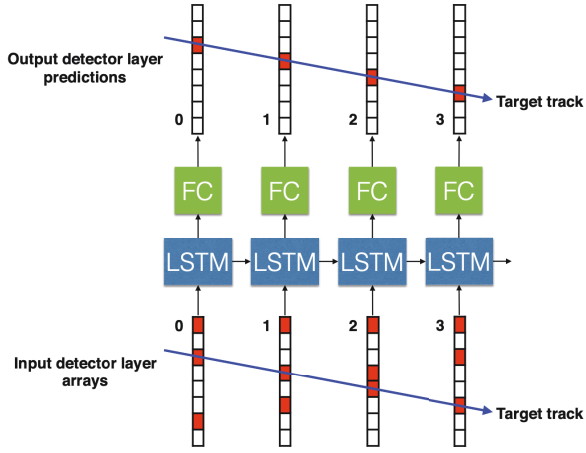


Figure 11.4: LSTM network taking in tracker layers as inputs and giving hits assigned to a track as an output. Figure from [112].

Another approach studied was to predict track parameters that describe the final track fit directly from the inputs, instead of assigning the hits and reconstructing the tracks. This would be an end-to-end model, taking raw detector information as input and producing a list of track parameters as an output. An example of such input and the sampled model output is shown in Figure 11.5. The uncertainties are achieved by requiring the network to predict the track parameter covariance matrix on top of the parameters. In this simplified example only track intercept and slope parameters were required to fully describe a track.

The results presented by the HEP.TrkX collaboration proved to be an interesting foray into more machine learning based track reconstruction. While the models presented were simplified compared to realistic track reconstruction, the study was a necessary proof of concept and made clear some difficulties that the algorithms should overcome to be viable. For example as the number of tracks in an event is not set, the models should be able to predict a flexible number of output tracks. The number of pixels per layer in the tracker subdetector is not fixed, so the model should be able to take variable length inputs as well if it uses the pixel arrays directly. For image-based approaches in realistic detector conditions the representation of information becomes sparse by design, since the tracking detectors are

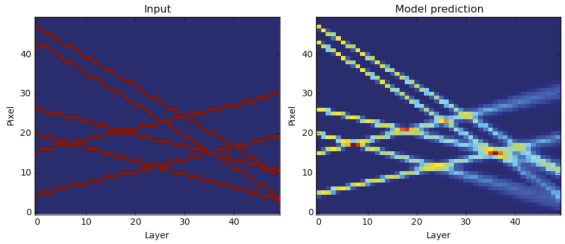


Figure 11.5: **Left:** 2D image input to a convolutional neural network that can be used to predict track parameters from the raw detector measurements. **Right:** Output produced by sampling the track parameters multiple time based on their covariance matrix. This gives an estimate of the uncertainties in the prediction. Figure from [112].

designed to have high granularity for accurate measurements and to avoid dead-time due to multiple particles activating the same pixels simultaneously. Sparse representations lead to inefficient learning as many computing operations are wasted on empty inputs and make the learning process more difficult. Regardless of these issues, the appealing feature in these methods is the linear scaling of computing requirements with respect to number of tracks in the event.

EXA.TrkX is the follow-up project based on the findings from HEP.TrkX. One of the driving considerations is to use methods that improve the representation of the input data to avoid sparseness. The chosen approach uses Graph Neural Networks (GNN) where the hit measurements in the tracker are the nodes of the graph and the task is to learn the edges connecting the hits of adjacent layers that belong to the same track. This represents the input data as a space-point cloud, where only valid hits and edges connecting them are considered instead of the whole pixel array. An illustration of this is show in Figure 11.6. The use of GNNs for tracking was first studied in [121] and based on these promising results of perfect or near perfect hit assignments to tracks in a simple tracking environment with a handful of tracks, this approach was studied further in [120] using a realistic HL-LHC tracking scenario an average of 200 pile-up vertices per event.

As an output the GNN provides likelihood score for each of the edges in the input. By choosing a threshold value the best suitable edges are chosen for track reconstruction. A simple algorithm iteratively visits all the hits from inner to outer layers reconstructing the best track candidate guided by the GNN output values. Each hit is only used for one track in this version of the algorithm. The latest published results are able to reconstruct 95% true tracks in the region studied. The current on-going work aims to improve the lost efficiency by improving the used track reconstruction algorithm.

This study shows the scalability of the GNN approach to track reconstruction with high potential to be usable in future tracking at the HL-LHC conditions. Even if the physics performance doesn't reach the currently used algorithms, it could be considered as an initial iteration of track reconstruction that can build a significant portion of the tracks fast and

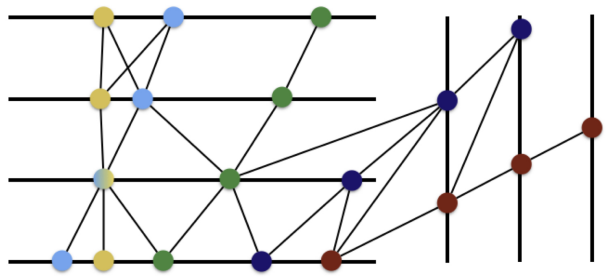


Figure 11.6: An illustration of how the input for a GNN can be formatted. The color of the hits represent the true track label they belong to while the edges are determined by loose geometrical constraints during preprocessing. Figure from [121].

which can be complimented with other more specialized algorithms that are able reconstruct the tracks the GNN misses.

Track building is the most work intensive step in track reconstruction as the current algorithms start from a track seed in the inner layers of the detector and search compatible hits from outer layers by propagating the estimate of the track flight path. One way to reduce the amount of computation is by only performing the track building on track seeds that actually correspond to a true track. For this purpose using convolutional neural networks to study classify track seeds is studied in [122].

The presented study is focused on the High-Level Trigger track seeding, but the same principles can be applied to the offline track seeding as well. Compatible pairs of hits in the innermost layers are formed into a hit doublet to be considered as a seed. Both hits are formatted as an image of using the charge distribution in the pixel detector region with the identified hit. As the charge distribution depends on the properties of the particle such as the energy, mass and its direction, comparing the charge distributions of two subsequent hits should contain the necessary information to see if they are compatible to have been caused by the same particle. Figure 11.7 shows the charge distributions of two hits in subsequent layers.

After a series of convolutional filters additional hit information is concatenated to the network as an auxiliary input. This input contains for example detector information and hit coordinates. The input is further processed by a series of fully connected layers and the final output layer returns a prediction if the hits are from the same particle or not.

Using $t\bar{t}$ events at $\sqrt{s} = 13$ TeV with average pile-up $< \mu >= 35$, the trained network is able to keep 99% of the formed hit pairs that correspond to an actual track while rejecting 85% of the pairs that do not. Since no computing resources have to be wasted on performing track building on the rejected track seeds being able to accurately choose the correct track seeds to build on can give a large time saving in track reconstruction.

It is worth noting that while the above discussion was largely from the CMS perspective other LHC experiments face similar issues and have also deployed neural network based tracking

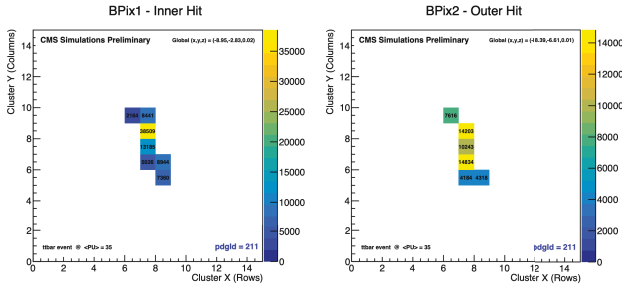


Figure 11.7: Distribution of measured charges between two subsequent layers of the pixel detector. Figure from [122].

solutions. Prominent examples are the fake track filter used in the LHCb experiment called the *ghost probability* algorithm [123] and the neural network based algorithm responsible for splitting charged clusters in the pixel detector of the ATLAS experiment [124] which improves tracking performance in environments with large particle density like the core regions of energetic jets.

11.5 Object reconstruction and identification

Similar object reconstruction tools used in the trigger-level are also useful for the offline reconstruction with the added benefit of not being under as strict time constraints. Machine learning methods are present in almost all object reconstruction and identification algorithms that are currently considered as state of the art.

Charged leptons: In the CMS experiment all of the charged leptons use machine learning at some point of their reconstruction or identification. For electrons a multivariate analysis (MVA) is performed to first select promising track seeds that are compatible with signals in the ECAL detector that are reconstructed into electron track candidates. An MVA regression is performed to determine energy corrections to the energy measurement of the electron, while another MVA regressor estimates the associated uncertainty to the correction. Finally a fourth MVA method identifies the electron candidate [125]. Reconstruction of tau leptons relies on Boosted Decision Trees (BDTs) to discriminate hadronic tau decays from jets by combining information on the isolation of charged tracks, the energy distribution of the particles and the life time of tau. Another BDT is used to discriminate tau particles decaying into electrons from isolated elecrons from other sources [126]. For offline muon reconstruction machine learning methods are not currently used, mostly due to the CMS detector having excellent performance on muon reconstruction using other algorithms [127].

Jets: A lot of interesting results using deep neural networks for object reconstruction have come from jet substructure based flavour taggers. This is understandable as there is a lot of high-dimensional but low-level variables that contain useful information for the task, but only

recently the deep learning tools have become more popular in the particle physics community. DeepJet [128] relied on CMS detector's Particle Flow candidates to extract particle-level information in addition of the jet-level variables to the jet flavour tagging algorithm using a combination of convolutional layers and Long-Short Term Memory layers to learn from the ordered sequence of particles contained in the jet. The new ParticleNet [129] uses a Dynamic Graph Convolutional Neural Network that takes an unordered list of particles assigned to the jet, similar to point clouds in computer vision tasks. There are also image based tools where the jet is formatted into a 2D image of energy deposits and the tagging is based on the learned distributions of these deposits [130]. An example of what an average W jet image after preprocessing looks like is shown in Figure 11.8, where the substructure of the two-pronged W decay is clearly visible. As jets are very much present in all hadron-hadron collisions, accurately identifying what caused the jet and reconstructing them for offline analysis has a large effect in the physics performance of the experiment. In addition to resolving which particle caused the jet, machine learning can be applied to correct for detector effects in the measurement of the energy as is shown in [131] for the case of b jets. There a neural network both predicts a correction to the jet energy measurement and the jet's resolution. An improvement of 12%-15% to the b jet resolution measurement is reported compared to baseline methods.

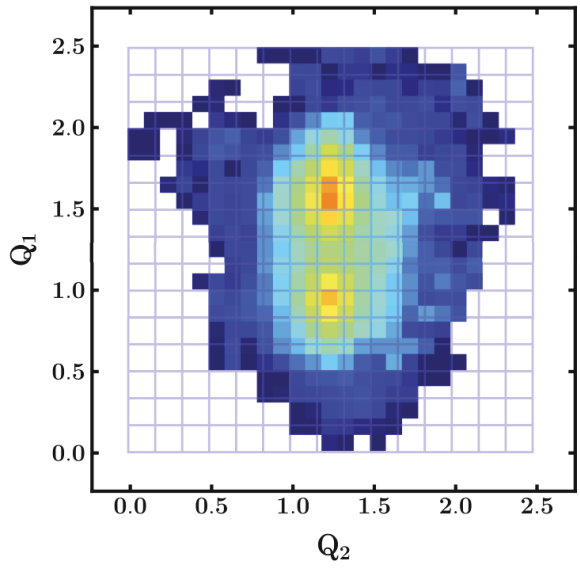


Figure 11.8: Average jet image for W jets with leading jet p_{T} between 200-250 GeV. The pixel color signifies the amount of energy deposited in the region. Figure from [130].

11.6 Offline analysis

After the collision event has been reconstructed to higher level objects that are calibrated with corrections to account for various measurement effects, the last part of the chain is performing an analysis where some quantity like a branching fraction to certain decay channel or number of events satisfying certain selections is determined and the result is compared to what the standard model predicts and possibly also to what other models of particle physics predict. In any case the analysis includes selecting some suitable subset of the collision events that contain as much of the event type that one tries to analyse as possible and have little contamination from other processes.

This event selection is done based on the reconstructed objects and their quantities. For example an analysis that is interested in measuring some property of the standard model Higgs boson will try to get as pure a sample of Higgs boson like events as possible. Any non-Higgs boson event will only serve to reduce the resolution of the measurement. To select a signal region usually it requires physical intuition about the process being studied, in this case one could choose only events where one can reconstruct a particle with an invariant mass within some mass window around the Higgs boson's known mass using the decay products of known Higgs boson decay channels. Additionally many other similar cuts improving the purity of the subset of events by removing background events that leave similar decay products in the detector are performed. Finding these selection cuts is often a combination of iterative testing and intuition to find the selections that maximize the sensitivity of the analysis. However this sort of multivariate analysis using a high-dimensional input to produce a classification to Higgs like and non-Higgs like events is something machine learning algorithms are thriving at, especially when there is simulation data with the known true labels available.

For this purpose the high energy physics community has been using various machine learning methods for a long time. The first use of feed forward neural networks in a particle physics analysis was at DELPHI in 1992 [132] where the Z boson's hadronic branching fractions to b and c quark pairs were measured. Also the Higgs boson discovery relied on multiple BDT classifiers for reconstructing the diphoton vertices in the $H \to \gamma\gamma$ decay channel [133].

Part III

Accelerators and detectors

Chapter 12

Large Hadron Collider

The Large Hadron Collider (LHC) is located in the countryside outside of the city of Geneva at the border of Switzerland and France, where also the CERN sites are located. The original plan for the accelerator was approved in 1994 and its construction spanned decade leading up to first beams accelerated with the machine in September 2008. The apparatus is housed in a 26,7 km long circular tunnel between 45 to 170 meters underground which was previously used for the Large Electron-Positron collider. It is both the largest machine in the world and the highest energy particle collider up to date.

The LHC was built to find physics beyond the Standard Model of particle physics with the center of mass energies up to 14 TeV. In order to achieve the highest possible energies, the machine was designed to accelerate and collide hadrons instead of leptons like it's predecessor had. Two beams of hadrons are made to circulate in opposing directions within the accelerator ring. In order to reach the high energies, the hadrons will go through a chain of preaccelerators before the LHC, as the accelerating radiofrequency (RF) cavities can only operate at certain design frequencies. After reaching the expected collision energies the beams can be directed to collide in some or all of the four interaction points around the ring, where the particle collider experiments that collect the data from the collisions are hosted. Due to the circular design of the collider, after reaching the designed collision energy the beams can be kept circulating while collisions take place every bunch crossing for hours with the detectors recording a constant stream of observations. The limit on running time with the same beams is mostly set by scattering of the proton bunches taking place during collisions. When the beam conditions are degraded up to some threshold, the beams are dumped and the machine is reset for the next beam fill.

The detailed technical documentation of the LHC can be found at [134]. In the following some of the technical details most relevant to the physics reach of the machine are presented, the operations so far are summarised and the future of the LHC is outlined as it is known at the time of writing.

12.1 Accelerator complex

The CERN accelerator complex including the experiments is presented in Figure 12.1. The need for so many accelerators comes from the RF cavities used to accelerate the charged hadrons, as they are only tuned for a certain frequency range corresponding to a certain energy range of the beam. Additionally in order for the particles to stay on the orbit with radius R, the strength of the magnetic field B in the accelerator for a particle with momentum $|\vec{p}|$ and charge q has to follow

$$B = \frac{|\vec{p}|}{qR}. \tag{12.1}$$

With the increasing energy and as such increasing momentum, the magnetic field strength has to keep growing to keep the particles in the orbit. And finally the amount of **synchrotron radiation** emitted by charge particles on a circular orbit will depend inversely on the square of the bending radius of the orbit so in order to limit both the energy loss and the radiation damage to the accelerators, a strategy of moving the accelerating particles to larger and larger circular accelerators as the energy increases is also needed to protect the accelerators.

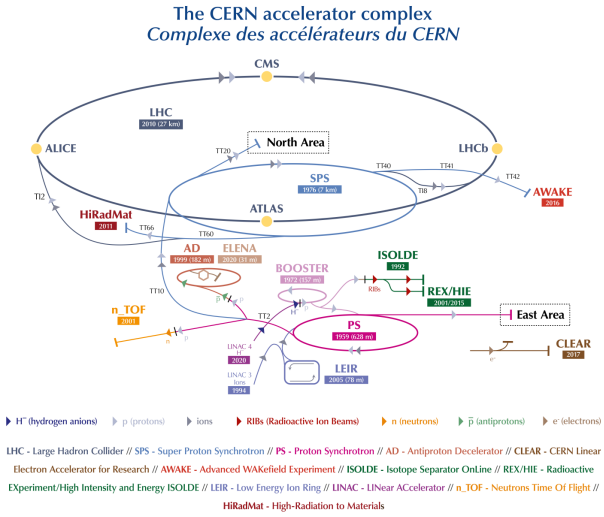


Figure 12.1: Illustration of the accelerators and the detectors at CERN. The LHC accelerator chain goes through the accelerators in the order LINAC2, Booster, Proton Syncrotron (PS), Super Proton Syncrotron (SPS) and LHC. Figure provided by CERN [135].

All the protons used are stored in a one relatively small bottle of hydrogen. The hydrogen atoms are **ionized** by stripping away their electrons with an electric field and the protons are then charged with a linear accelerator LINAC2 up to the energy of 50 MeV. The beam gets passed onto the Booster where they are accelerated to the energy of 1.4 GeV. From Booster

the protons transfer to the Proton Syncrotron to be accelerated to 25 GeV and after that to the Super Proton Synchrotron where the energy is increased up to 450 GeV. With 450 GeV energy the beam can be submitted to the LHC ring which can continue the acceleration all the way up to the design energy of 7 TeV per beam, although during Run 2 the used energy was 6.5 TeV to ensure the machine's safety. Table 12.1 shows these energies and the fraction of the speed of light this corresponds to for the protons.

Accelerator	Energy	Fraction of speed of light
LINAC2	50 MeV	0.314
Booster	1.4 GeV	0.916
PS	25 GeV	0.9993
SPS	450 GeV	0.999998
LHC	6.5 TeV	0.99999999

Table 12.1: The energies of the beam after the accelerator.

As can be seen from Table 12.1, the absolute velocity of the protons does not seem to increase much after reaching the energy of 25 GeV at the PS. This is due to the relativistic effects that become important at such high velocities, storing the additional energy gained by the particles into their masses. However for particle collisions the important factor is the available collision energy or the **center-of-mass** energy of the colliding particles.

12.2 Center-of-mass energy

As the mass-energy relation declares $E = mc^2$, where E is energy, m is mass and c denotes the speed of light. In particle collisions, the E would usually signify the center-of-mass energy of the two particles colliding so the proton-proton system at the LHC. The E gives the maximum rest mass of the particle that can be created in the collision through energy becoming mass. The center-of-mass energy $\sqrt{s} = p_1 + p_2$ where p_1 and p_2 are the four-momenta of the colliding particles.

When accelerating elementary particles such as electrons and positrons, the particles participating in the collisions are each carrying the energy they have been accelerated to. With composite particles such as protons that consist of quarks and gluons it is more complicated, since the collision is actually between the elementary components of the protons interacting with each other and these particles will only have a fraction of the energy of the whole proton. How the energy is distributed between the elementary particles making up the proton is described by the **parton distribution functions** but for now it suffices to say that it is very unlikely for the whole energy of the proton to be carried by any single parton and as such the center-of-mass energy in hadron-hadron collisions is unlikely to be the sum energy of the two composite particles being accelerated.

The LHC is designed to reach as high a center-of-mass energy as possible with the technology that was considered achievable at the time of designing it. For this reason the accelerator uses protons instead of electrons for example, since the synchrotron radiation scales inversely to the fourth power of the mass of the particle and the amount of radiation from electrons

accelerated to the LHC collision energies would add tremendous difficulties to keeping the superconducting magnets operational. Figure 12.2 demonstrates why high center-of-mass energies are desirable as it shows production **cross sections** for different Standard Model processes as a function of the center-of-mass $\sqrt{s}$. Cross section gives the likelihood of a process taking place. It is usually given in barns which is a unit of area. For purposes of high energy physics barn is a large quantity so nano-, pico- or femtobarns are more commonly encountered.

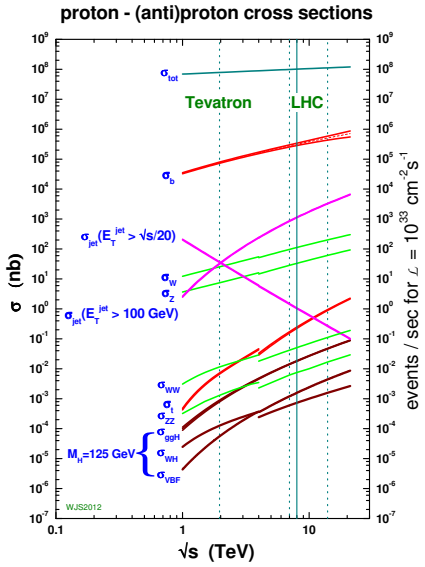


Figure 12.2: Cross sections for various Standard Model processes as a function of the center-of-mass energy. There is a discontinuity at 4 TeV where due to switch from proton-antiproton collisions at the Tevatron to proton-proton collisions at the LHC at that energy. Image from [136].

Notably the right-hand scale in Figure 12.2 shows how many events per second are expected at a fixed instantaneous luminosity, demonstrating how one needs to run the collider less time to get a desired number of collisions event of certain type if the center-of-mass energy is higher. Also worth noting how different the scales of the total hadronic cross section σ_{tot} is compared to for example the largest Higgs boson production cross section σ_{ggH}. At the LHC energies these are $\mathcal{O}(10^8)$ nb and $\mathcal{O}(10^{-1})$ nb respectively so there is a huge amount of other collision events for every single Higgs boson event taking place. This is why searching the Higgs boson was said to be like finding a needle in the haystack and in order to make any meaningful physics analysis on the processes with smaller cross sections, one has to device clever triggering and selection schemes that are able remove as many of the uninteresting events without losing the signal. Even with high energies many processes like the Higgs production are very rare. In order to accumulate enough of the rare events to claim something like a discovery of a new particle, **instantaneous luminosity** of the collider is an important design aspects as well.

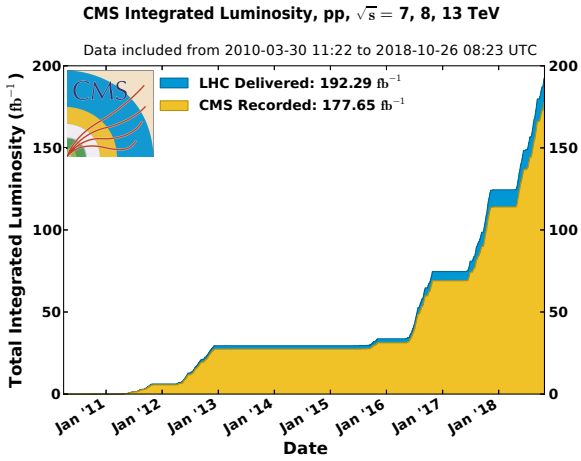


Figure 12.3: The integrated luminosity delivered by the LHC and collected by the CMS detector for Runs 1 and 2. Figure from [137]

12.3 Luminosity

The number of events per unit time i.e. the **production rate** can be determined using the likelihood of the process which is the cross section and the instantaneous luminosity $\mathcal{L}$ of the collider:

$$\frac{dN}{dt} = \sigma\mathcal{L}. \tag{12.2}$$

Instantaneous luminosity is in the units of $\text{cm}^{-2}\cdot\text{s}^{-1}$. For the two high luminosity experiments at the LHC, ATLAS and CMS, the target peak instantaneous luminosity is 10^{34} $\text{cm}^{-2}\cdot\text{s}^{-1}$ To get the total number of events collected over some period of time T, the production rate can be integrated over time to get the **integrated luminosity** L:

$$N = \sigma \int_0^T \mathcal{L}dt = \sigma L. \tag{12.3}$$

Integrated luminosity is often used as a measure of the amount of data collected, for example the CMS detector is said to have collected 160 fb^{-1} of data during data taking of Run 2. Figure 12.3 shows the amount of data delivered by the LHC and collected by the CMS detector for both Run 1 and Run 2.

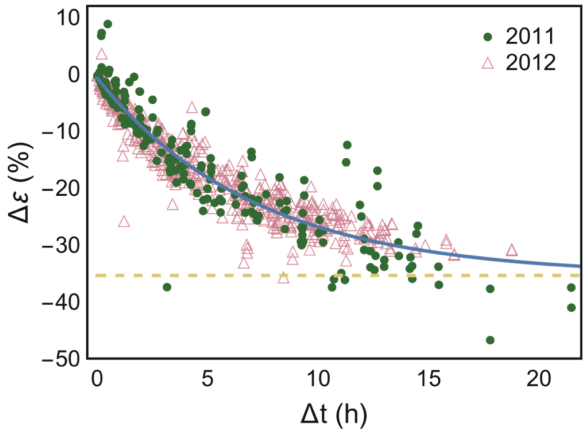


Figure 12.4: Relative change in instantaneous luminosity between the start and end of a fill due to beam degradation during operations. Particles are lost due to beam-beam interactions during bunch crossings. Figure from [139].

As more luminosity means more data and as such larger statistics for observing new physics processes, instantaneous luminosity is one variable collider design tries to maximize. This can be done by increasing the likelihood of protons colliding in a bunch crossing by squeezing the protons to a more point-like region when colliding and manipulating the crossing angles or by increasing the frequency of bunch crossings taking place. These depend on the magnets used in the accelerator.

A pair of beams is called a **fill** and during the lifetime of a fill the instantaneous luminosity delivered by the LHC decreases as a function of time. This is largely due to degradation of the circulating beams due to collisions. Not only will the protons participating in a hard collision be lost from the beams, but also other protons that might only slightly interact during a crossing which causes them to drift away from the proton bunch. The protons that are at risk of drifting so far that they might hit the beam pipe walls are cleaned out by specific magnets in the LHC. This is done in order to protect the superconducting magnets from heating up due to protons colliding with the magnet. The evolution of the beam during collisions is studied in [138, 139] and the difference in instantaneous luminosity as a function of time between the beginning and the end of a fill is presented in Figure 12.4.

The protons inside the beams are not organized as a continuous stream but instead form **bunches** of protons that are typically approximately 10 cm long. These bunches are separated from each other by 25 ns when the bunches are moving at speeds just a fraction below the speed of light. This 25 ns separation also determines the frequency of collisions taking place at the interaction points and it corresponds to 40 million collisions per second or 40 MHz collision frequency.

12.4 Magnets and RF cavities

The LHC relies on powerful superconducting magnets to manipulate the charged particles in the collider while superconducting RF cavities are used to accelerate the particle bunches and keep the beams at the target collision energy. The maximum energy of the beams is determined by the nominal magnetic fields of the dipole magnets that bend the trajectories of the protons to stay within the accelerator circle. The LHC dipole magnets operate at 8.33 T, which corresponds to top energy of 7 TeV for the protons. Achieving such magnetic fields requires cooling the magnets down to only a few Kelvins temperature where the niobium-titanium (NbTi) alloy is superconducting, passing through electric current without any resistance. The magnets are cooled slightly below the temperature required for NbTi to become a superconductor, since effects like the synchrotron radiation can deposit energy into the magnets during the acceleration and heat up the material. If a magnet heats up too much it can transition into a non-superconductive state causing resistance to the large currents inside the magnets. This resistance will start heating up the magnet further and can cause significant damage to the magnet. Such a sudden loss of superconductivity in the magnet is called **quenching**. The temperature margin between the operating temperature and the critical temperature of the superconductor in the LHC magnets is relatively small, requiring tight control over any sources of heating to the magnets during beam operations.

The design decision to build the LHC into the existing LEP tunnel with 3.7 m diameter limits the possibility of building two individual rings for the two beams. This led to the use of a "two-in-one" superconducting magnet design where both beams are circulating in the same magnetic ring, but have separate beam pipes [140]. A cross section of a dipole magnet used at the LHC is shown in Figure 12.5, where this design can be clearly seen. While this was more cost and space efficient, it couples the rings magnetically reducing flexibility when operating the machine.

The LHC contains various other types of magnets than the dipoles as well, with specific functions varying from squeezing the proton bunches to smaller transverse cross sections before the interaction points to cleaning particles straying too far from the beam pipe center out of the bunches to prevent the risk of them colliding to the beam pipe walls. The LHC has more than 50 different types of magnets and around ten thousand superconducting magnets in total are needed for the machine to operate. They are cooled using 120 tonnes of liquid helium and an electric current of 11 kA is needed to produce the 8.3 T magnetic field.

There are 16 RF cavities in the accelerator ring that are responsible for first accelerating the bunches to the LHC collision energy and then maintaining the bunches at that energy even when energy is being lost through the synchrotron radiation. The RF cavities have a strong electric field of 5 MV/m. This electric field oscillates at the frequency of 400 MHz so that a proton with exactly the correct collision energy will not experience a net force when passing through the RF cavity. A proton with a lower (higher) energy will be accelerated (decelerated) towards the right energy due to being out of sync with the electric field oscillation in the cavity.

Inside the beam pipe the particles travel in a vacuum in order to avoid nuclear scattering of protons on the residual gas. The design requirements set the level of vacuum in the beam

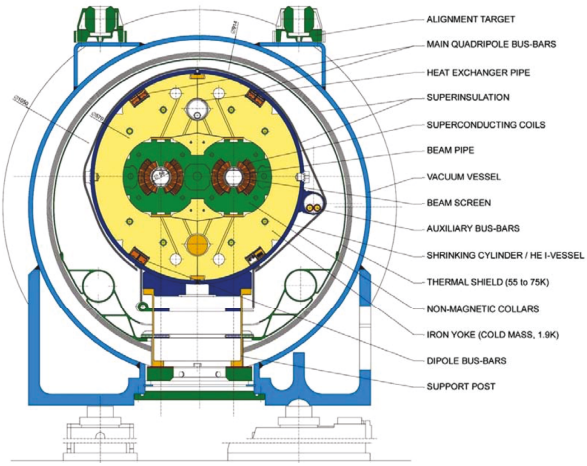


Figure 12.5: The "two-in-one" magnet design used at the LHC, where the two beam pipes are at the center. The superconducting coils of NbTi are surrounded by various cooling and structural support materials in order to keep the alloy at a superconducting state and hold the magnet together against the significant mechanical stress caused by the strong magnetic field during operation. Figure from [134].

pipe to a level that allows for 100 hour beam lifetime before the beam-gas collisions decay the beam intensity too much. This also prevents the magnets from heating up too much due to scattered protons dispersing energy to the magnet materials. The vacuum inside the beam pipes is comparable to the particle densities of the interstellar void with pressure of 1.013 $\cdot 10^{-13}$ mbar. Additionally the cooled superconducting magnets and the helium distribution lines are insulated using vacua but these do not have as stringent requirements as the beam pipe vacuum.

12.5 Achievements so far

The LHC has so far completed two periods of data taking called Run 1 (2009-2013) and Run 2 (2015-2018) where a total of 168 fb^{-1} of proton-proton collision data was delivered to the CMS and ATLAS detectors. Additionally datasets of lead-lead, lead-proton and xenon-xenon collisions studying the quark-gluon plasma state of matter have been produced with the LHC.

During Run 1 the LHC had already produced enough data that an observation of a new Higgs boson like particle could be announced on 4th July 2012 by the CMS and ATLAS collaborations [141]. The results of the CMS collaboration in the $H \rightarrow \gamma\gamma$ are shown in Figure 12.6, where the measured data is shown with the expected background fit if there was no Higgs boson in the displayed mass range. Later analyses with more data have shown the observed particle to be so far consistent with the Standard Model Higgs boson in many of its

properties, but due to its elusive nature there are new tests and studies becoming available as the amount of collected data as well as the analysis methods improve. For example the $H \rightarrow b\bar{b}$ decay was only recently confirmed [127, 142] even though the Standard Model predicts a branching ratio of 58% into $b\bar{b}$ for a 125 GeV Higgs boson, making it the dominant decay mode of the Higgs boson.

While the LHC has not produced evidence of any Beyond Standard Model theories such as supersymmetry, thousands of research papers have been published using the data produced by the LHC with the CMS experiment nearing its thousandth publication at the time of writing. The data during the first two runs have been used in e.g. observing new penta-quark states [143], measuring the top quark mass at an unprecedented accuracy [144] and observation of jet quenching due to formation of quark-gluon plasma at the LHC [145]. However after over a decade of operations, the LHC has produced only a small fraction of the collisions that will be seen during its lifetime.

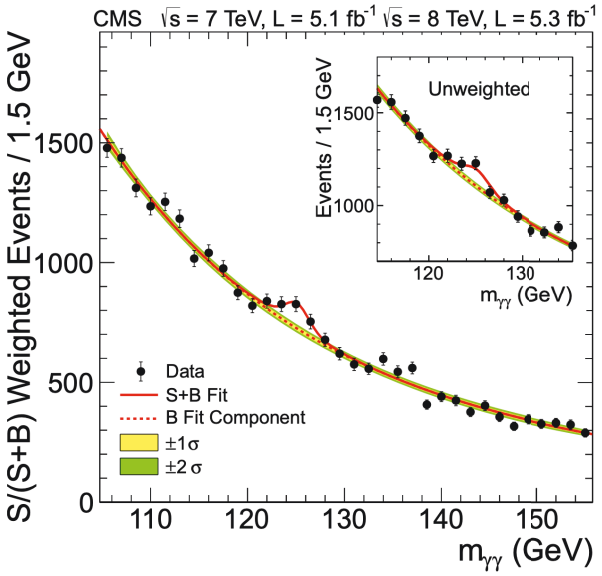


Figure 12.6: The discovery plot of the Higgs boson decaying into two photons from the CMS collaboration [8]. The diphoton channel provided a clean experimental signal in the detector and was the first one where a significant excess of events compared to the background hypothesis was discovered, even though it is not the dominant decay channel of the Standard Model Higgs boson.

12.6 The future of the LHC

The LHC will start Run 3 of data taking in May 2021 and continue until the end of 2024. After Run 3 the LHC will be upgraded during a two and a half year long shutdown in order to

increase the instantaneous luminosity of the collider by a factor of five compared to the LHC design value. This new collider is called High Luminosity Large Hadron Collider (HL-LHC) and it will operate in the same tunnel as the LHC. The HL-LHC project is presented in [146]. The increased luminosity will allow more accurate analysis of rare processes like the Higgs boson production and the possibility of seeing even weaker signals that have not yet been detected. In the current plans the HL-LHC will be operational until the end of 2036. The physics prospects of the HL-LHC for the CMS and ATLAS collaborations have been collected in [147].

The HL-LHC design requires some technologies that do not yet exists but are deemed achievable, including magnets capable of 11-12 T fields during operation and new technologies for beam collimation. The goal of the HL-LHC is to produce 3000 fb^{-1} of data during its operations, compared to the 300fb^{-1} goal of the LHC. This requires extensive upgrades in the LHC accelerator chain. Additionally the new HL-LHC conditions will subject the detectors to even harsher conditions than before, requiring substantial upgrades in both hardware and software. Additionally after the three runs of the LHC, some of the subdetectors are due to be changed in any case due to degradation from the high radiation environment over the years. Especially the increased pile-up in the collisions will prove to be a challenge in order to avoid degrading the detector performance. The detectors will need to prepare for pile-up of $< \mu >= 200$ compared to the $< \mu >= 32$ experienced by the CMS detector during 2018 proton-proton collisions. This will add requirements to the algorithms used in the event reconstruction from the detector outputs as well as the physical properties of the used detectors.

Chapter 13

Compact Muon Solenoid experiment

The Compact Muon Solenoid (CMS) is stationed at IP5 at the LHC ring, in the middle of the French countryside. It is one of the two general-purpose high luminosity experiments at the collider, studying various topics from Higgs boson to dark matter and supersymmetry. The CMS collaboration is made out of over 4000 scientists of various backgrounds representing more than 40 countries around the world.

13.1 The Compact Muon Solenoid

The CMS detector is composed of multiple subdetectors in an onion like structure. The CMS detector size and structure with the different subdetectors is illustrated in Figure 13.1. While the compactness of the detector that is the size of a four story building can be argued either way, characteristic to the CMS detector is its strong superconducting solenoid generating a magnetic field of 3.8 T during operations and its superb efficiency and accuracy in reconstructing muons are what give the detector it's name. The particles are set to collide in the center of the detector and the products from that collision will propagate outwards through the detector layers. The subdetectors of the CMS are the **silicon tracker** measuring trajectories of charged particles, **electromagnetic calorimeter** (ECAL) that stops and measures photons and electrons, hadron calorimeter (HCAL) that showers and absorbs hadronic collision products and the **muon chambers** that detect muons punching through the other detector layers. Additionally the ECAL has a **preshower** detector in the endcap regions and the forward regions next to the beam pipe are covered by a **forward calorimeter**.

Combining the signals from different layers, different particle types can be identified and their properties measured. The detector is designed to be **hermetic** i.e. to surround the interaction point and prevent particles from escaping. Only the weakly interacting neutrinos escape the detector volume unseen, but their energies and directions in the transverse plane can be inferred from the **missing transverse energy** $\vec{p}_{\mathrm{T}}^{\,\mathrm{miss}}$ after reconstructing all the

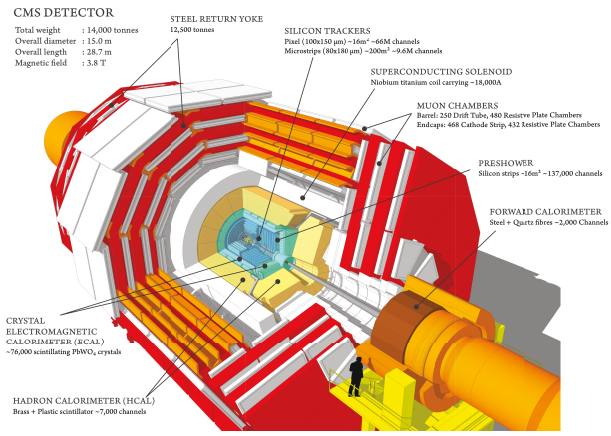


Figure 13.1: An illustration of the CMS detector design displaying the separate subdetectors and other structures making up the CMS detector. A human figure for scale is shown at the lower center of the image. Image from [148]

other particles and taking a vector sum of their four-momenta. As the initial four-momenta in the transverse plane when the collision takes place is zero, due to laws of conservation any net four-momenta after summing up all the other particles should come from neutrinos. In practice of course misreconstruction and mismeasurement of other particles causes small errors in the neutrino momenta determination.

Using the information of different subdetectors to identify the various particle types is illustrated in Figure 13.2. For example the photons are reconstructed by finding energy deposits in the ECAL that do not have a corresponding charged particle track pointing towards them in the tracker. The HCAL aims to stop every particle, but muons are able to push through and are the ones leaving signals in the muon chambers at the outer edge of the detector.

The design goals for the CMS detector can be summarized as

- High performance muon identification and momentum resolution over a large momentum range, good charge determination of muons and good dimuon system mass resolution.
- Accurate tracker that is able to resolve charged particle momenta with high reconstruction efficiency near the interaction region at the heart of the detector as these tracks are needed for τ and b jet tagging.
- Good electromagnetic energy resolution.
- Good missing transverse energy and dijet mass resolutions that require a hermetic hadron calorimeter coverage.

These requirements mostly determine the ordering of the subdetectors. The silicon tracker has to be close to the interaction region for tagging secondary vertices needed in b jet and τ

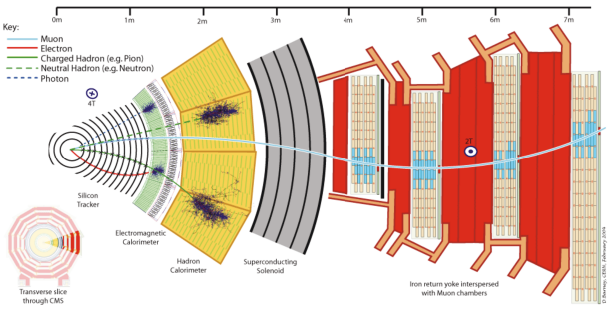


Figure 13.2: A section of the CMS detector cross section in the transverse plane. Signals left by different types of particles are illustrated in the figure, demonstrating how the identification proceeds by taking advantage combining the information of multiple subdetectors. Image from [149].

identification. In order to determine the charges of leptons, a strong magnetic field is needed but the solenoid has to be outside the calorimeters in order not to interfere with the energy measurements of ECAL and HCAL. Since muons and neutrinos are the only particles passing through the whole detector, muon chambers at the outer edges of the detector can be used to improve muon identification and and energy resolution.

The CMS experiment uses a right-handed coordinate system where the origin is at the nominal collision point, the y-axis points vertically upwards and the x-axis points radially inwards towards the center of the LHC ring. The z-axis is aligned with the beam pipe direction. The azimuthal angle ϕ is measured from the x-axis in the x-y plane and the radial distance in this plane is denoted r. The momentum in the $x-y$ plane is denoted as the transverse momentum $\vec{p_{\mathrm{T}}}$. Polar angle θ is measured from the z-axis. A coordinate often used in describing the particle collisions is the *pseudorapidity* η defined in terms of the polar coordinate θ

$$\eta = -\ln\tan(\theta/2). \tag{13.1}$$

13.2 Tracker

The goal of the CMS tracker is to provide high precision three dimensional point measurements along the trajectories of charged particles propagating through the detector. The detector contains multiple layers of silicon detector material and read-out electronics with the necessary cooling and powering systems. The inner layers called the **pixel detector** provide a finely grained measurement system of initially 66 million individual pixels of 100 μm by 150 μm in size. This ensures low enough occupancy in each pixel to be able to distinguish between separate tracks and provides high quality measurements that can be used as a starting point when reconstructing the tracks of charged particles. The large instanta-

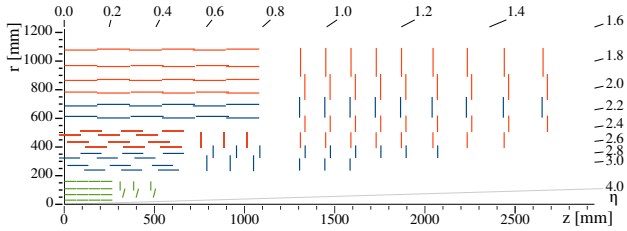


Figure 13.3: One quarter of the Phase1 CMS tracker in r-z view. Pixel detector, one-sided strip modules and stereo strip modules are depicted in different colors. Figure from [150].

neous luminosity of the LHC exposes the pixel detector to significant radiation that degrades the performance of the measurement device over time. Additionally the larger luminosities translate to more charged particles increasing the occupancy of the pixels in the detector as multiple particles risk hitting the same portion of the detector nearly simultaneously. The hit rate in 2016 data taking was 40% over the pixel detector design goal, requiring an upgrade to the detector after Run 1. During this upgrade the pixel detector was replaced and a new pixel layer in both the barrel and the endcap were added increasing the total number of pixels in the system to 124 million.

Following the pixel detector layers are the coarser **strip layers** made of 9.6 million strips with their width varying between 80 μm and 205 μm. In order to provide two-dimensional coordinates for measurements, part of the strip layers are made out of stereo modules, where two strip sensors are mounted back-to-back and aligned at a 100 mrad relative angle.

The tracker can detect charged particles up to $|\eta| < 2.5$ with the layers arranged into a cylindrical shape around the interaction point. The four pixel barrel layers are positioned between 2.9 cm and 16.9 cm away from the interaction point radially along r in the transverse plane and the three forward disks between 3.2 cm and 4.8 cm from the interaction point in z along the beam pipe. The ten strip layers in the barrel region span the distance 25 cm to 110 cm in the radial coordinate and the 12 strip endcap layers go up to 280 cm in the z. The positioning of different tracker layers are shown in Figure 13.3.

The tracker is based on the charged particles depositing energy in the silicon wafer as they pass through. This energy can ionize electrons from silicon atoms creating electron-hole pairs in the material. With a voltage applied across the wafer, the charges can be collected at edges of the silicon and read out as an electric signal. Due to the strong magnetic field inside the tracker, the charged particle trajectories are bent. This allows the measurement of the particle's charge and momentum using the coordinates where a particle has deposited it's energy in subsequent layers of the tracker.

13.3 Electron calorimeter

The electron calorimeter (ECAL) at the CMS detector was designed with the search for the Standard Model Higgs boson decaying into two photons in mind. This $H \to \gamma\gamma$ decay channel has a relatively low branching ratio of around 0.002, but the signal produces a

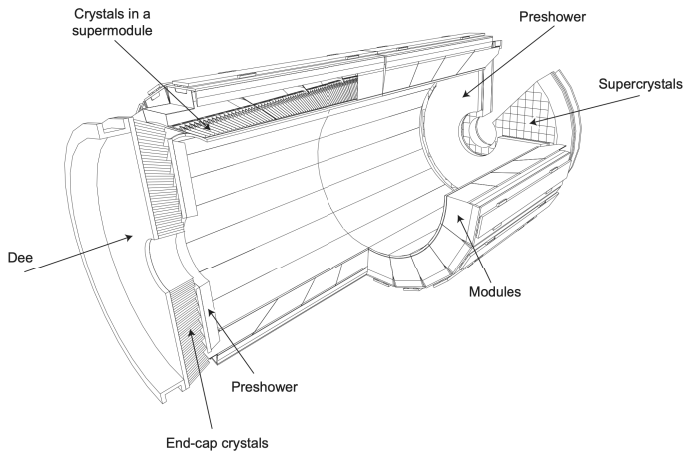


Figure 13.4: Schematic picture of the ECAL detector. In the barrel region (BE) of the cylinder shaped detector, endcaps (EE) composed of two Dees each and preshower (ES) component that improves the EE sensitivity. Figure from [152]

narrow resonance peak over a continuous non-resonant diphoton background making this a clean channel for the Higgs boson detection. This has proven to be an excellent choice for finding and measuring the Higgs boson found at the LHC. [124, 151]

The ECAL detector had to conform into having excellent energy, position and angle resolutions up to tracker coverage of $|\eta| < 2.5$, large dynamic range of measuring energies between 5 GeV and 5 TeV, being compact and hermetic, having a fast response time of 25 ns in order to be used in trigger decisions, and being able to withstand significant amount of radiation without suffering from degraded performance. Structurally the ECAL detector consists of barrel (BE) and endcap (EE) regions. Additionally in front of EE there is a preshowering detector (ES), which helps reduce diphoton background resulting from the decay of π^0. The ECAL is near hermetic extending up to $|\eta| = 3.0$. Its setup is visualized in Figure 13.4.

Main feature of the ECAL detector is the array of lead-tungsten crystals ($PbWO_4$), used to absorb and re-emit the energy of photons and charged particles through scintillation. This material is dense, allowing the 23 cm (22 cm) long crystals to have 26 (25) radiation lengths worth of material in the BE (EE). In order to have high granularity required for the excellent spatial resolution, the frontal face of the crystals is 2.2 cm$\times$2.2 cm (2.7 cm$\times$2.7 cm) in the BE (EE). It also has the required fast signal production capability, as it emits 80% of the absorbed energy within 25 ns. The downsides of this compound include the strong dependence between light yield and the operating temperature and the relatively low light yield in general. This necessitates both the strict operational temperature constraints requiring powerful cooling systems to be installed and additional on-board amplifiers to the readout systems. These crystals are organized into 36 supermodules in the EB, each supermodule containing 1700 crystals. In the EE the elements are grouped into supercrystals of 5×5 crystals and a small number of special shaped supercrystals near the inner and outer edge of the EE.

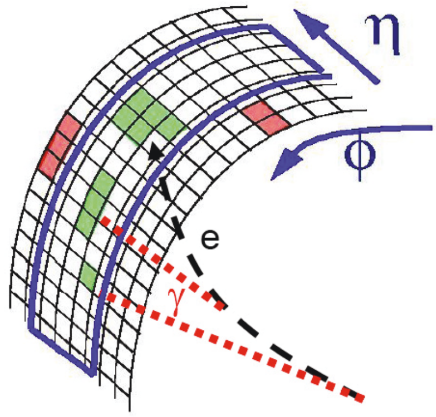


Figure 13.5: Charged particles can emit their energy via bremsstrahlung before arriving at the ECAL. In order to determine the original energy of the particle, these stripe shaped energy deposit patterns need to be recognized and summed together, in a procedure called dynamic clustering. Figure from [153]

The energy from a showered particle tends to spread out over multiple crystals. Additionally bremsstrahlung from a charged particle may already be emitted before arriving at the ECAL, leading energy deposits from photons being detected in stripes in the $\eta - \phi$ plane in ECAL. This effect is depicted in Figure 13.5. In order to get the energy of the original particle the deposits of multiple crystals need to be added together for the actual measurement. This is called clustering.

While ECAL has performed exceedingly well during Run 1 and Run 2, the radiation damage to the detector requires constant monitoring and additional corrections to the measurements. In 13.6, the dependence of relative response $R = E_{\text{measured}}/E_{\text{true}}$ over different dates during Run 1 are depicted. Here the effect of degrading detector conditions is clear and it can be seen that the more radiated parts of the detector at high values of $|\eta|$ suffer more from this effect. This loss of response is largely attributed to the $PbWO_4$ crystals darkening due to defects caused by hadronic radiation to the material. This darkening reduces the amount of emitted photons reaching the readout electronics, effectively resulting in energy loss in the measurements. Using the ECAL laser monitoring system [154] these effects can be quantified and the measurements can be recalibrated to take this into account.

More detailed description of the ECAL can be found in the CMS detector description [156] and the performance of the detector during Run 2 is presented in [157].

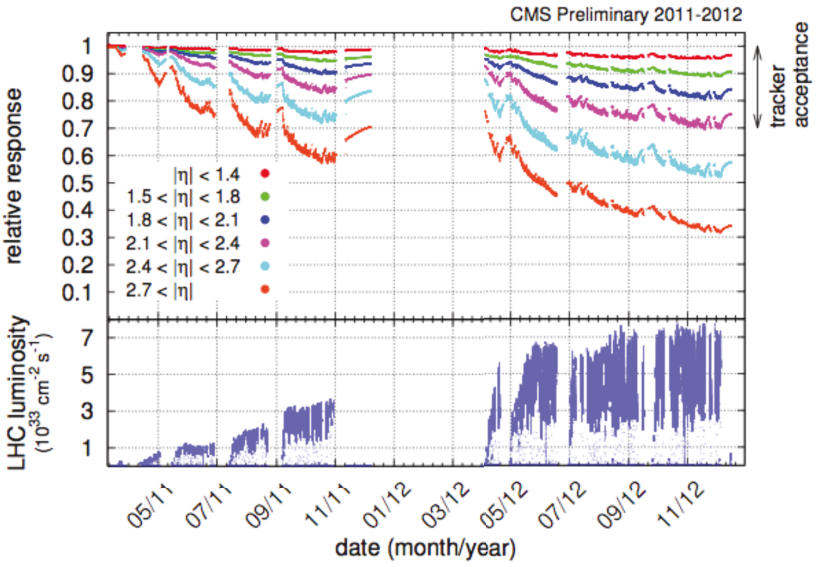


Figure 13.6: Radiation damage during operations decreases the response of the detector. The measurements are done with the built in ECAL laser monitoring system, that can be used to accurately recalibrate the ECAL measurements. The relative response is the ratio of the measured energy and the true energy of the particle. Figure from [155]

13.4 Hadron calorimeter

The Hadron Calorimeter (HCAL) is the next subdetector moving away from the interaction point. Its purpose is to force hadronic particles to interact with its dense absorber material and start showering. Structurally it is built by interleaving active scintillating material with plates of absorber material, so that based on the signal from the scintillator the energy of the showering particle can be reconstructed using detailed models of hadron interactions with material.

Restrictions in designing the HCAL mostly come from the limited space and the need to be able to stop the particles within the calorimeter volume to get an accurate energy measurement. This requires the use of dense absorber materials to initiate nuclear interactions between the detector and the particles. In order to provide measurements, the design approach of the HCAL was to interleave dense passive absorber material with thin layers of active detector material. These interactions with the absorber cause **hadronic showers** in the detector which are measured with scintillator plastic. To determine the full energy of the particle including the portion lost in the absorber, modelling of the hadronic showers is used with the input from the scintillators to reconstruct the energy. Additionally since the particles arriving at HCAL have already traversed through a significant amount of material, some of them may have deposited energy already in the ECAL. This is taken into account in the reconstruction, where measurements from both the HCAL and the ECAL are combined.

HCAL consists of four different regions: barrel (HB), endcaps (EB), outer layer (HO) and the forward calorimeters (HF). The barrel and the endcap regions are made out of brass plates as absorbers and a scintillating plastic as the active material. These cover the $|\eta|$ ranges between 0.0-1.4 and 1.3-3.0 respectively. In order to add more depth in terms of interaction lengths the HB is complemented by HO, providing additional scintillator layers and a tail catcher iron for the central region outside the magnetic coil. Roughly 5% of hadrons with $p_{\mathrm{T}} \geq 100$ GeV leave signal in the HO. To make the detector as hermetic as possible, a forward detector is added to fill the $|\eta|$ range from 3.0 to 5.2. This very forward region is an extremely radiated area, requiring the detector to withstand harsh conditions. For this reason the HF is made out of steel plates with quartz fiber material detecting the Cherenkov radiation from charged particles. These features are collected in Table 13.1

The HB and HE active layers are grouped into towers and the measured light from the towers are added up to produce the final signal. In HE and at the far edges of the HB, the towers are split into two or three sections that are individually added together to get a depthwise splitting of the signal. The HO scintillators are grouped into five rings each covering multiple towers. This structure is displayed in Figure 13.7. The towers and the rings shown in the slice are organized as 36 wedges around the beampipe.

In total when reaching the outer edge of any part of the HCAL detector system, the particles will have traversed roughly 10 hadronic interaction lengths through the detector meaning that almost all the hadrons should leave a signal to the calorimeters and only very few push through to the outer layers of the detector in what is called a **punch-through event**. More details on the HCAL detector can be found at [156] and on its performance at [158].

part	$\|\eta\|$	details
Barrel (HB)	0.0-1.4	Scintillator plastic interleaved with brass absorber, located between the EB outer edge and the solenoid magnet
Endcap (HE)	1.3-3.0	Scintillator plastic interleaved with brass absorber. Located behind ES and EE.
Outer (HO)	0.0-1.2	Inner ring at $\|\eta\| \leq 0.35$ has an additional iron plate absorber between two scintillators. Elsewhere consists just from a scintillator plastic layer without additional absorber. Located outside the solenoid magnet.
Forward (HF)	3.0-5.2	Due to extreme radiation in the forward region, quartz fibers used as the active medium. Detects Cherenkov radiation from charged particles. Steel plates as absorber.
Magnet coil	0.0-1.4	No active material, acts as an additional absorber for HO. Located outside the HB.

Table 13.1: Sections of the HCAL system and their special features.

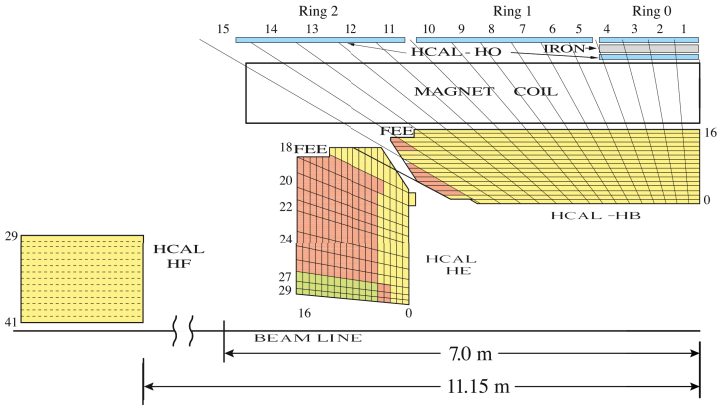


Figure 13.7: A quarter slice of the HCAL detector structure. Here the HB, HE, HO and HF sections of the detector and the magnetic coil are displayed. The detector is segmented into layers of scintillating material, with the same colored elements of each tower being added optically for readout. FEE shows the location of Front End Electronics for readout. Figure from [158].

13.5 Muon detectors

The muon detector system was a central theme in designing the CMS detector as the name implies. This stems from the standard model Higgs boson's predicted decay channel $H \rightarrow ZZ/ZZ^* \rightarrow llll$. For the case where the four leptons resulting from this decay are all muons this would provide an excellent mass resolutions for the Higgs boson measurement as long as the muons were reconstructed accurately.

The CMS muon system is located at the outer edges of the detector. This positioning allows it to have a relatively low occupancy due to almost all other particles except having been stopped in the calorimeters. Additionally this provides a longer arm for determining the charges and the momenta of the detected muons based on the curvature of their trajectories in the CMS magnetic field.

There are three different types of muon detectors in the CMS for satisfying different requirements based on their location: Drift Tubes (DT) in the barrel ($|\eta| < 1.2$), Cathode Strip Chambers in the endcap ($0.9 \leq |\eta| \leq 2.4$) and Resistive Plate Chambers in the barrel and intermediate region ($|\eta| < 1.7$). While the technology used in each of the different detector type varies they all are able to produce a location and timing measurement of a muon passing through. This timing information is important for being able to use the measurements of the three subsystems that correspond to the same bunch crossing i.e. a 25 ns time window. Without timing the muon detections from subsequent crossings might get mixed up in the reconstruction process. The layout of the muon detector is depicted in Figure 13.8.

In the barrel region the DTs are made of tubes filled with ionizable gas and a positively charged wire strung across in the center. When a charged particle like a muon crosses through the tube, it ionizes the gas and the resulting negatively charged electrons drift to the center wire in the electric field. These electrons can be measured as an electric signal from the wire, signifying that a particle has passed through. These DTs are positioned in five layers around the barrel so that part of them run parallel to the beam pipe and part of them orthogonally. This allows for the determination of the point where the muon crosses each layer.

In the endcaps the CSCs are able to operate with high rates and in non-uniform strong magnetic fields. They are robust and do not require especially careful gas, temperature or pressure control. While the physical operating principle is similar to DT in that the electrons from the ionized gas drift to a wire and cause an electrical signal, the CSC is made out of an array of positively charged anode wires and negatively charged cathode strips. Additionally by setting the strips and the wires perpendicular to each other, one can use the signal from electrons flocking to the wire as one coordinate and the positive ions going to the strip as another coordinate effectively getting a 2D measurement from one CSC. The close spacing of the wires and strips also make CSCs fast detectors suitable for triggering.

The RPC detectors are made out of two highly resistive plates positioned close to each other with ionizable gas in between. One of the plates is a positively charged anode and the other is a negatively charged cathode. The cathode is equipped with external metallic strips that are able to pick up the charge of the electrons resulting from ionization of the gas when a muon

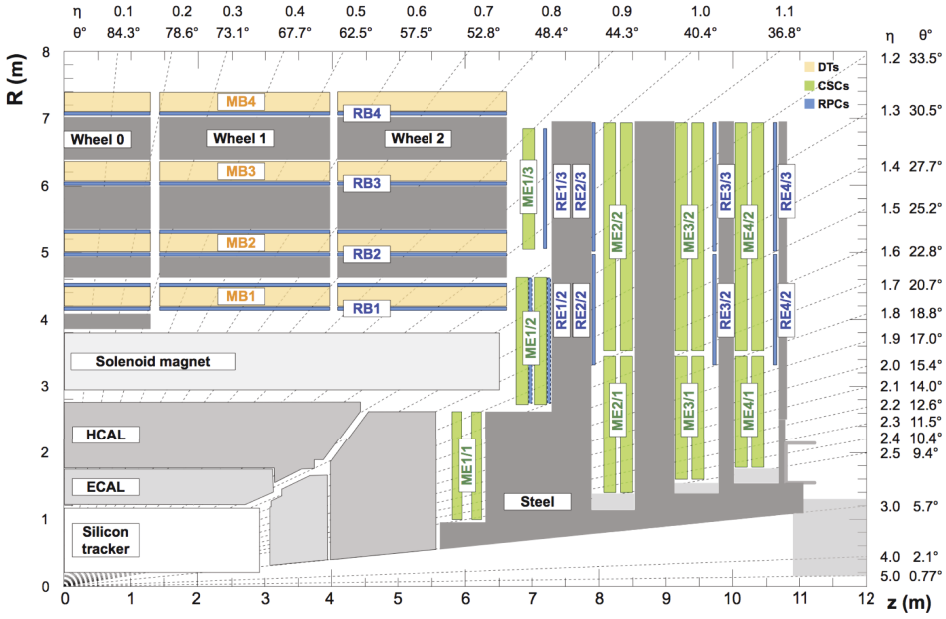


Figure 13.8: A quarter slice depicting the CMS muon detector. Different detector types are color coded in the image. Figure from [159].

passes through. The amount of charge in the strips and their neighbourhood can be used to interpolate where the muon crossed and estimate its momentum. The RPC provides a fast time resolution of the order of a nanosecond combined with good spatial resolution.

The information from these three different types of detectors are combined in order to get the information from muons in the collisions fast and precisely. The muon system is used as one of the important trigger systems that can quickly decide if a collisions contained anything interesting that causes high energy muons and if it should be stored for further analysis. During Run 2 the CMS muon system enabled a greater than 96% efficiency in reconstructing and identifying muons [159]. A more complete documentation of the CMS muon system can be found in [156].

13.6 Trigger system

During data taking the proton bunches collide inside the CMS detector every 25 ns which corresponds to a collision rate of 40 MHz. While the detector itself is designed with this in mind, the amount of data produced by the sensors if every collision were to be recorded is infeasible for modern storage technologies. This leads to the dilemma of choosing which events are worth saving and which are not, and how to tell them apart.

The trigger system at the CMS experiment is responsible for selecting which collisions to store for further analysis. Since the collisions are taking place 40 million times per second these decisions must be made roughly at the same rate. Fortunately only a small portion of the collision events are of particular interest to the physicists like the ones producing Higgs bosons or other heavy particles that might be sensitive to signs of new physics while the rest might be considered an unavoidable background of well studied and established Standard Model processes that take place when protons are being collided. The task of separating the events of interest from the rest has been split into two steps in the CMS detector: Hardware trigger called Level 1 (L1T) and a software trigger called the High Level Trigger (HLT). This two tiered approach allows the first trigger to make the rapid fire decisions on whether the collision can be discarded instantly or if it might be worth keeping. The L1T is allowed to accept events at the rate of 100 kHz and pass them forwards to the HLT for more complex analysis. The HLT runs a coarse version of the CMS event reconstruction where some of the charged particle tracks get reconstructed and kinematic parameters of high level objects like the muons can be calculated to make a more informed decision if the event should be stored. The HLT is allowed to accept events at the rate of 1 kHz and pass them onto the data storage to wait for offline analysis.

Trigger has a crucial role to play in a particle physics experiment as it has the sole power over accepting or rejecting events before anyone has the change to analyse the collisions thoroughly and the rejected data will not be retrievable once the decision is made. Additionally different physicists may have different ideas of what types of collisions are the most important ones leading to strenuous negotiations over how the available bandwidth for storing collision data should be designated i.e. what should the trigger choose to keep.

L1T: In order to cope with the large data stream, dedicated integrated circuits called field-

programmable gate arrays (FPGAs) are used instead of general purpose Central Processing Units (CPUs) that an ordinary computer would have. The FPGAs can be configured to specifically perform their designated task without the overhead of having to support a general purpose instruction sets and operations that have been designed with other applications in mind. The L1T receives inputs from a subset of the CMS subdetectors: ECAL, HCAL and the muon system. L1T runs simplified reconstruction algorithms targeted at performing a coarse energy reconstruction for energetic particles and energy sums such as computing the missing transverse energy or the transverse energy of a jet. Based on these reconstructed variables, trigger decisions are made by usually requiring particular objects like muons to be present in the trigger reconstruction and having momenta over a predetermined trigger thresholds. By increasing or decreasing these thresholds, the amount of accepted events can be tuned to match the available bandwidth.

HLT: Events accepted by the L1T get passed onto the HLT processor farm where the full event data from the detector is processed to provide a HLT reconstruction of the event. The main principle is the same as for L1T, where the events are required to contain some objects of interest that satisfy set kinematic restrictions. The HLT gets to also use tracker information and perform a fast version of the CMS track reconstruction algorithm. This is useful for tasks like tagging possible b-quarks in the events as secondary vertices can be detected from reconstructed pixel track information. The HLT combines the information from different subdetectors using a simplified version of the Particle Flow (PF) algorithm presented in Section 14.1 in the context of offline event reconstruction. In order to satisfy the computing time constraints, the HLT aims to filter out the events as fast as possible, by discarding the whole event once the first filtering condition is fulfilled. This allows the PF algorithm to be fully run only on a small subset of the events passed to the HLT.

The full description of the CMS trigger system and its performance during Run 1 can be found at [160]. Summary of the upgrades the trigger system went through for Run 2 can be found at [161].

Chapter 14

Event reconstruction

The full capability of the layered structure of the CMS detector is realized when the information from all the subdetector systems are combined to perform the reconstruction of the collision event. The CMS event reconstruction aims at global event description where every particle in the collision is accounted for. The hermetic design of the detector allows the signatures of weakly interacting particles like neutrinos be reconstructed indirectly by first reconstructing all the other particles and then assigning the missing momenta in the event over to these particles. The process responsible for the CMS event reconstruction is called the particle-flow (PF) algorithm detailed in [162]. Here a brief description of the process is provided.

A crucial component for this approach is the high performance tracker at the heart of the detector that allow the energies measured in the coarser calorimeters to be assigned to the right particles. Additionally this approach allows cross-calibrating different subdetectors, increase accuracy of measurements by using multiple measurements of the energy by different portions of the detectors, and reducing detector background by identifying and masking them.

14.1 Particle flow

The PF algorithm is split into three steps: Reconstructing tracks and vertices, clustering and calibrating calorimeter energy deposits, and linking together the tracks, calorimeter clusters and muon chambers which can then be classified into different types of PF candidates.

Track reconstruction takes the information from the silicon sensors in the tracker and clusters them to hits. These hits describe the point where a particle passed through the detector and combining the points from subsequent layers a track representing the trajectory of a charged particle can be formed. In order to achieve high efficiency while maintaining low rate of mistakes leading to reconstructing so called fake tracks, the track reconstruction algorithm is applied iteratively while using different layers and kinematic constraints to produce the initial track seed that is used to build the track. Track reconstruction and especially the task of further reducing the misreconstruction rate in the track classification step is discussed

in detail in Chapter 17.

Vertex reconstruction is the process of determining the location and associated uncertainty for the locations where charged particles in the event were created. For primary vertices this means finding all the proton-proton interactions. The process is done by first selecting good quality tracks to use for vertex finding, clustering the tracks together based on whether they seem like they originate from the same interaction, and using tracks that are clustered together to fit the vertex location. In order to choose the correct tracks, the impact parameters that describe the closest approach between the track and the beam spot are constrained, a number of pixel and strip hits are required to be associated with the tracks and the maximum value of the normalized χ^2 from the fit to the trajectory is limited.

The selected tracks are grouped together based on their impact parameter in the z-coordinate along the beam line. Assigning different tracks to different vertices is done using a deterministic annealing algorithm [163], which finds the global minimum for the optimization problem similarly to a physical system that settles at its minimum energy through gradual temperature reduction.

After grouping the tracks the vertices with at least two tracks pointing at them are fitted using an adaptive vertex fitter [164], which computes the estimate for vertex coordinates and their uncertainties. The number of degrees of freedom computed using the adaptive vertex fitter has a strong correlation with the number of tracks that are coming from the interaction region, so the vertex n_{dof} can be used as a qualifier for selecting the true proton-proton interaction vertices among all of the reconstructed vertices. The CMS efficiency on reconstructing primary vertices is high especially when three or more tracks are associated with it when it reaches efficiencies of $> 99.5\%$, as can be seen in Figure 14.1 for Run 1 conditions. The measurement for data is done using a tag and probe method where a fraction of the tracks clustered by their z_0 impact parameter are randomly split into two sets that are independently fit, one of them representing the truth (tag) and one of them being the test set (probe). The efficiency is calculated based on the number of times the probe vertex is reconstructed and matches the tag vertex.

Calorimeter clustering and calibration aim to retain a high efficiency for event the low-energy particles reaching the calorimeters and of being able to separate close energy deposits. For this purpose an algorithm based on a Gaussian mixture model (GMM) fitted using maximum-likelihood is used on the aggregated calorimeter cluster seeds after zero-suppressing the sensors to remove noise. The GMM assumes the energy deposits of the cluster to be caused by N sources distributing their energy around them following a Gaussian distribution. The parameter N is chosen as the number of calorimeter cells where the energy deposit exceeds a predetermined seed threshold. So the N Gaussians in the mixture model represent the different particles that may have hit the calorimeter in the same sector so that their energy deposits in the cluster overlap. The parameters of the GMM are iteratively fitted until convergence, after which the locations of the different Gaussians in the model are used as estimates for the positions of the particles when they reach the calorimeter.

Careful calibration of the calorimeters is performed using radioactive sources, test beams and cosmic rays. For the ECAL in particular, the detector response is evolving through time as

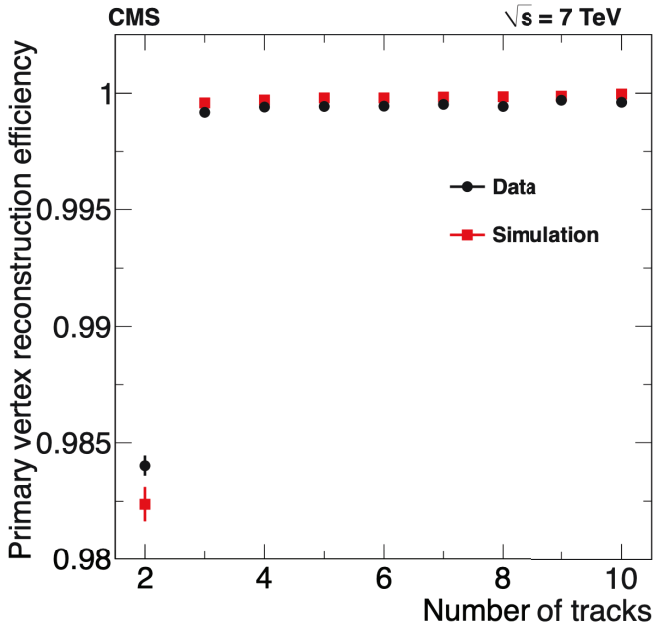


Figure 14.1: Primary vertex reconstruction effficiency with respect to number of tracks assigned to the vertex using Run 1 minimum-bias data and MC simulation. The efficiency quickly raises to nearly 100% when there are three or more tracks associated with the vertex. The data and simulation are in good agreement. Figure from [165].

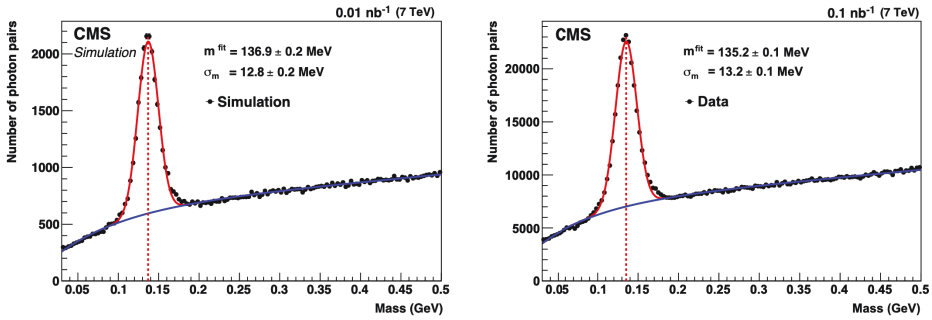


Figure 14.2: Invariant masses of photon pairs in the $|\eta| < 1.0$ region for simulation (left) and data (right). π^0 signal is shown as the Gaussian red curve over the exponential background in blue. Figure from [162].

the radiation damage accumulates darkening the lead-tungsten crystals used as scintillator material. The corrections are derived using GEANT4 detector simulations [166]. An analytical correction function in η and E is fitted to the two dimensional distribution to determine the corrections. The function tends towards unity at high energies, but at lower energies the corrections can be as large as +20% in the barrel and +40% in the end cap regions. These simulation driven corrections can be verified using the abundant π^0 samples decaying into photon pairs in the collected data from proton-proton collisions. The percent level agreement between reconstructed π^0 invariant masses in simulation and data demonstrated in Figure 14.2 validates the method used for deriving the ECAL energy corrections.

Hadron energies are distributed both in the ECAL and the ECAL cells. For this purpose another function of pseudorapidity η and energy E that combines the energies measured in the ECAL and HCAL clusters is determined for the hadron calibration and fitted using simulated single neutral hadrons. The response, i.e. the mean relative difference between the measured energy and true energy of the particle, is shown alongside with the resolution for both the raw and calibrated measurements in Figure 14.3. The response and resolution are shown as a function of the true energy of the particle. A significant improvement is seen in the calibrated response, demonstrating the necessity of this correction to hadrons due to the energy being spread into ECAL as well as HCAL.

Linking the signals reconstructed in the tracker, calorimeters and the muon chambers allows for the full picture of a particle to be reconstructed. For charged particles, the reconstructed tracks can be extrapolated outwards to the other elements in the outer layers of the detector and matched to any compatible signals in the calorimeters. Tracks originating from a secondary decay of some short-lived particle are linked together. For particles like electrons where an abrupt change in trajectory due to radiated photons is possible, specialized algorithms are used to improve the track fit. Signals in the muon detectors are matched with tracks to produce the full muon trajectories in the detector. To prevent the linking algorithm from testing any pair of elements in the detector which would lead to a computational time that grows quadratically, the linking algorithm is restricted to consider only the nearest

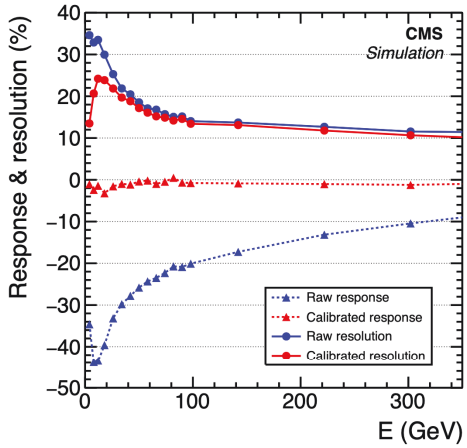


Figure 14.3: The response and resolution for the raw and calibrated measurements as a function of the true particle energy. Values closer to 0 in the y-axis are better. The improvement produced by the calibration are significant, as mismeasurements of up to 40% are visible at low energies for the raw response. Figure from [162].

neighbours in the (η, ϕ)-plane

14.2 Identification and reconstruction

Each PF block represent an object or multiple objects to be reconstructed and identified. This process is done in a sequential manner, removing the information associated with an object that is already reconstructed from consideration of the next reconstruction steps in the sequence.

Muons get reconstructed first using the signals from the muon systems and the tracker that have been assigned to the PF block. Muons where there is a muon track that can be built from signals in the muon chambers as well as a track that can be linked to it in the tracker are known as global muons. These are the most accurately reconstructed ones due to the additional information from two independent subdetectors that can be used for the measurement. The global muons are supplemented by standalone muons that are built by using only the muon chamber signals and tracker muons that are built by extrapolating a track to the muon systems and linking it with at least one DT or CSC sensor segment.

Electrons and isolated photons are reconstructed with the tracker and calorimeter information in the PF block. Due to the large fraction of energy the electrons can lose through radiation and the possibility of these radiated photons undergo the pair production process into e^+e^--pairs, the reconstruction and identification of photons and electrons is treated with a common approach. Electrons can be seeded with an energy deposit in the ECAL or by a reconstructed track linked to an energy deposit in the ECAL. BDT algorithm that has been trained to identify electrons based on up to fourteen variables including the amount of radiated energy, number of hits in the track and goodness-of-fit is used to do the final classification of PF candidates as electrons. Photons are seeded by energy deposits in the ECAL that have not been linked with a electron track candidate.

Hadrons and nonisolated photons can be identified once the signals linked to muons, electrons and isolated photons are removed after they have been reconstructed. The remaining signals are the results of hadrons from jet fragmentation and hadronization. These are charged hadrons ($\pi^{\pm}$, $K^{\pm}$ or protons), neutral hadrons (such as K^0_L or neutrons) and nonisolated photons that can result from for example π^0 decay.

Energy clusters in ECAL and HCAL that are not linked to any track are interpreted as photons and neutral hadrons. Within the tracker acceptance $|\eta| < 2.5$, all ECAL the clusters are reconstructed as photons and all HCAL clusters as neutral hadrons. Although this can falsely assign some neutral hadron energy into photons, within hadronic jets 25% of the energy is in photons that will be stopped at the ECAL while the neutral hadrons leave only 3% of their energies to the ECAL so the approximation is well justified. Outside of the tracker acceptance in the forward regions, the signals from charged and neutral hadrons are not separable due to lack of tracks, so the reconstruction assigns all ECAL clusters without a linked HCAL cluster into photons and all HCAL clusters with or without a linked ECAL cluster get assigned to hadrons.

As an additional refinement for the calorimeter clusters linked to tracks the energy measurements from both the calorimeters and the tracker can be compared. If the calorimeter energy exceeds the track energy, additional energy can be assigned as photons and/or neutral

hadrons. If the calorimeter energy is smaller than the track energy, additional muons that were originally hidden inside the jets will be searched for in the PF block with relaxed muon reconstruction constraints.

Missing transverse energy is the indirect signal for weakly interacting particles such as the neutrinos. The quantity $\vec{p}_{\mathrm{T}}^{\,\mathrm{miss}}$ is the negative vector sum of the transverse momenta of all the particles reconstructed in the collision, with a small additional correction made to the energies of particles assigned into a jet to account for detector response and jet energy corrections

$$\vec{p}_{\mathrm{T}}^{\,\mathrm{miss}} = -\sum_{i=1}^{N_{\mathrm{particles}}} \vec{p}_{\mathrm{T},i} - \sum_{j=1}^{N_{\mathrm{PF\ jets}}} (\vec{p}_{\mathrm{T},j}^{\,\mathrm{corr}} - \vec{p}_{\mathrm{T},j}), \tag{14.1}$$

where each jet with the raw momentum of $\vec{p}_{\mathrm{T},j} > 10$ GeV is replaced with its corrected value $\vec{p}_{\mathrm{T},j}^{\,\mathrm{corr}}$.

These provide steps provide the backbone of the CMS PF algorithm that enables performing high precision analyses using the reconstructed global event where all the particles in the detector are available as individual objects. Additional refinements are made by identifying and calibrating higher level objects based on the reconstructed PF candidates. For example jets are identified as tight clusters of PF candidates in some portion of the detector using the anti-k_{t} clustering algorithm [167] and their energies get corrected for different types of detector effects that are caused by a large number of particles hitting the detector in a very concentrated region at the same time. The details for jet energy corrections are documented in [168].

Part IV

CMS Track Classification

In the CMS experiment the inner parts of the detector are reserved for detecting signals from charged particles with in intense radiation and high rate conditions. This is done using a silicon based tracker that consists of pixel detector and strip detector as detailed in Section 13.2. The multilayer structure allows for multiple measurements to be made for each charged particle depositing energy in the tracker system, which can then be collected together and reconstructed as the flight path of the particle called track.

However the process of reconstructing a track starting from the initial energy deposits detected in the tracker material requires multiple stages of processing the information and combining the signals from different layers of the tracker. This operation is generally referred to as tracking and it will be presented in detail for the CMS detector during Run 2 in this part.

The author has participated in the work of the CMS Tracking Physics Object Group throughout his thesis work, being responsible for validation of offline track reconstruction, upkeep and retraining of the machine learning based track classification algorithms and developing new improved algorithms for track classification.

Chapter 15

Track reconstruction

Track reconstruction at the CMS experiment can be split into five stages of processing the data provided by the readout electronics of the tracking detector:

1. Hit reconstruction
2. Track seeding
3. Track finding
4. Track fitting
5. Track selection

First the signals in the detector are processed and group together as hits that provide an accurate point in space where the particle has crossed a sensor in the detector. These hits are then used to reconstruct stubs of tracks by combining together compatible hits that could have originated from the same charged particle. Based on the momentum calculated from the track stub, the track finding step extrapolates the trajectory of the particle in the CMS detector magnetic field and assigns new hits encountered along this extrapolation to the track. After a collection of hits believed to originate from the same particle is determined a fit is performed using all the assigned hits to produce a track candidate. Finally this track candidate is classified as a high quality, regular or fake track based on parameters extracted from the track fit. After the last step of this process, a collection of reconstructed tracks is produced that can be used in other parts of the CMS event reconstruction chain.

15.1 Hit reconstruction

Pixel detector: The signals from individual pixels are zero-suppressed so that the threshold is set to require an individual pixel containing an equivalent charge of 3200 electrons in order to be counted as a signal. The activated pixels are then clustered by joining adjacent pixels (side-to-side and corner-to-corner) and the cluster of pixels is required to have at least 4000 electrons worth of equivalent charge. A minimum ionizing particle is depositing around 21000 electrons in the tracker material, so the threshold has ample margin for not ignoring actual

signals. The pixel detector uses two different approaches to the hit reconstruction, a fast algorithm used in track seeding step and a cluster shape based algorithm in the track fitting step for improved performance at a higher computational cost.

The fast algorithm known as the **first-pass hit reconstruction** determines the position of the cluster in the local coordinates *u, v* of the sensor element by projecting the cluster onto each coordinate in turn, by summing together the charge in the pixels. Based on the projected charges, the center of the cluster is determined by (using u coordinate as an example):

$$u_{\text{hit}} = u_{\text{geom}} + \frac{Q^u_{\text{last}} - Q^u_{\text{first}}}{2(Q^u_{\text{last}} + Q^u_{\text{first}})} \left| W^u - W^u_{\text{inner}} \right| - \frac{L_u}{2}, \tag{15.1}$$

where u_{geom} is the arithmetic mean u coordinate of the projected cluster, Q_{first} and Q_{last} signify the charges collected in the first and last pixels of the projected cluster. The last term $L_u/2 = D \tan \Theta^u_L/2$, which accounts for the Lorentz shift along the u-axis as the magnetic field imparts a force on the charge carriers with D being the sensor thickness and Θ^u_L the Lorentz angle in this direction. The factor $|W^u - W^u_{\text{inner}}|$ accounts for the fact that the two pixels at the edges of the projected cluster are not expected to be fully covered by the deposited charge. W^u_{inner} is the geometrical width of the projected cluster where the first and last pixels are excluded and W^u is the charge width that signifies the expected width for the deposited charged based on the angle α^u between the track and the sensor

$$W^u = D \left| \tan(\alpha^u - \pi/2) + \tan \Theta^u_L \right|. \tag{15.2}$$

To summarize Equation 15.1, it provides the local coordinate for the projected cluster in the sensor by correcting the arithmetic mean coordinate based on the Lorentz drift of the charges and how the charge is distributed in the edges of the cluster.

The more precise algorithm called the **template-based hit reconstruction** is used to counteract the effects of radiation degradation in the pixel detectors during their lifetime. As the fast algorithm above only used the first and last pixels of the projected cluster in determining the coordinate of the cluster it is highly sensitive for noise caused by heavy radiation damage in singular pixels. The template-based approach uses the full cluster charge distribution that is observed and compares it with expected projected distributions called the templates.

Templates for this purpose are generated using a physically motivated and detailed Pixelav simulation [169–171] that describes the interactions between the silicon pixels and charged particles. It can also account for radiation effects in the sensors, so new templates can be generated during the lifetime of the detector to keep the templates accurate. Since the angle between the track and the sensor affects how the charge is distributed, sets of templates are

prepared for several ranges of the crossing angle.

In order to choose a right template to describe the observed charge distribution, the projected cluster is compared with the templates of the corresponding crossing angle and computing the χ^2 statistic describing the goodness-of-fit using the number of charges in the pixels. Using P_i as the observed charge in pixel i, the simulated expected charge distribution $S_{i,j}$ in pixel i with index j denoting a finer binning of the charges within the pixel extracted from simulation and ΔP_i as the expected root mean square value of the charge P_i also extracted from simulation, the χ^2 for each template is calculated as:

$$\chi^2(j) = \sum_i \left(\frac{P_i - N_j S_{i,j}}{\Delta P_i} \right)^2 \tag{15.3}$$

where N_j is the normalization factor between observed charge and the template charge,

$$N_j = \sum_i \frac{P_i}{(\Delta P_i)^2} \Big/ \sum_i \frac{S_{i,j}}{(\Delta P_i)^2}. \tag{15.4}$$

As the simulation allows supersampling the charge distribution within a single pixel of the detector, the templates are able to provide finer granularity into determining the position where the trajectory of the particle crossed the sensor. The best precision is achieved by summing over all template bins j to find the best fit, different versions of this minimization can be done based on the restrictions in the computing budget. Additionally an uncertainty on the hit position can be derived by running the reconstruction algorithm over the samples used to generate the templates, since true hit positions are known in the simulation and any bias can be determined by comparing the reconstructed and true hit positions.

Strip detector: Similarly to the pixel detector a zero-suppression algorithm is run on the strip sensors to remove noise from the measurements. The strip hits are clustered with any strip exceeding its expected noise by a factor of three. Neighbouring strips are added to the cluster if their charge exceed the expected noise by a factor of two. A cluster is discarded if its total charge is less than five times the cluster noise, defined as $\sigma_{\text{cluster}} = \sqrt{\sum_i \sigma_i}$ where σ_i is the noise for strip i in the cluster.

The hit position is determined by the charge weighted average of the strip positions in the cluster, with a correction of 10 μm (20 μm) in the TIB (TOB) that accounts for the Lorentz drift. An additional correction term of 10 μm is added for the thicker 500 μm silicon wafers due to an inefficiency in collecting the charges generated near the back-plane of the sensitive volume of the silicon. This is a result from the narrow time window during which the readout chip integrates the collected charge and it causes the barycenter of the cluster to shift along the direction perpendicular to the sensor plane.

Uncertainty of the strip hit positions is determined using the charge width defined in 15.2, which depends on the crossing angle between the trajectory of the particle and the sensor. For the rare cases when the observed cluster width exceeds the expected charge width by a factor of 3.5 or more, a 'binary resolution' is used as the uncertainty estimate i.e. the width of the cluster divided by $\sqrt{12}$.

Based on Run 1 when known defective modules of the tracker are excluded, the achieved hit reconstruction efficiency is over 99% for both the pixel and the strip trackers. The efficiencies per module are shown in Figure 15.1 The hit resolution in the pixel detector is approximately 10 μm in the $r\phi$ -coordinates and between 20 to 45 μm along the z-coordinate depending on the incident angle between the particle and the normal plane of the silicon detector.

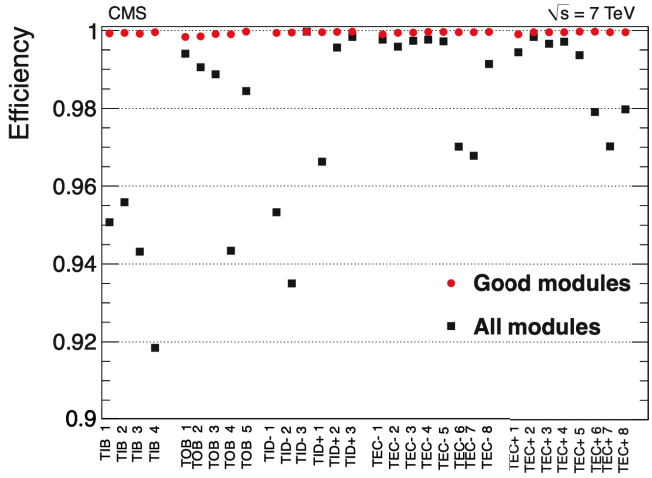


Figure 15.1: The hit reconstruction efficiency for different modules of the tracker. The red dots show the efficiency when known defective modules are excluded and the black dots show the total reconstruction efficiency. Figure from [165].

15.2 Track seeding

After the detector signals are reconstructed as hits, initial track seeds are produced by finding pairs, triplets or quadruplets of hits that are compatible for being originated from the same particle. These form the starting point of the track building algorithm as they provide an estimate for the necessary parameters in order to extrapolate the trajectory of the particle in the CMS detector. Inside the magnetic field of the tracker, charged particles follow a helical path that can be defined using five parameters. To uniquely determine these parameters either three 3-D position measurements or two 3-D position measurements and a constraint on the origin of the trajectory are needed.

To restrict the combinatorics of the track building task, a set of constraints is used when

selecting which hits can form a seed. First seeding layers are selected to determine which layers can produce acceptable hits for seeding. Tracking region specify limits of acceptable track parameters the seed has to satisfy, like the minimum p_T or maximum distances of closest approach to the assumed point of production of the particle. As the track building operation is computationally expensive, effort is put into trying to optimize the seed selection so that only seeds corresponding to actual particles would be built and sent forwards to track building step. This has a trade-off with the track reconstruction efficiency as only the tracks that have been seeded can be built.

A powerful tool for selecting the correct seeds is by comparing the charge distributions of hits in subsequent layers of the seed. As the trajectory is curved in the magnetic field, the crossing angles of the particle and different layers are correlated. Additionally the energy of the particle affects the ionization it causes in the silicon which will add another correlated variable to the production of charge distributions. Seeds can be categorized based on their seeding layers:

Pixel triplets are produced from three hits in the pixel layers. Tracks leaving such signature are called prompt tracks as they are produced by particles decaying promptly near the beam spot. Due to three measurements from the high granularity pixel layers, pixel triplets produce high quality track seeds.

Pixel quadruplets are produced with four hits in the pixel layers. Similarly to the triplets these seeds offer very high quality estimates on the track parameters due to the excellent performance of the pixel sensors.

Mixed pairs use an additional constraint from the vertex location. If more than one vertex is reconstructed, all of them are considered in turn for the track seed.

Mixed triplets require three hits formed from a combination of pixel and matched strip hits. Matched strips are built from the pixel modules that have two sensors mounted back-to-back so a 3-D position measurement can be determined from them. A mixed triplet seed has to have at least one pixel hit. The beam spot constraint is more relaxed to enable higher efficiency for tracks resulting from hadron decays, photon conversions and nuclear interactions. These are called displaced tracks as opposed to the prompt tracks.

Strip pairs are built with two matched hits from the strip detectors. The constraint on the point of origin is relaxed a lot. Strip pairs are used for reconstructing tracks produced outside of the pixel detector.

For Run 2 the track seeding algorithm was upgraded in the form of Cellular Automaton (CA) [172]. The standard method for finding pixel triplet seeds consisted of taking a pair of hits in the inner layers of the detectors and propagate the expected trajectory to the next layer to see if it would connect with a third hit. The CA algorithm hit triplets and quadruplets by starting with finding doublets that share a hit between them. This is depicted in Figure 15.2. The redesign in the seeding algorithm allowed for improved exploitation of the parallelism in the problem and also better data locality. As the compatibility for additional hits needs to be checked only for pairs of doublets that share a hit between them, the combinatorics in high rate environments is naturally limited to only the local environment of the track.

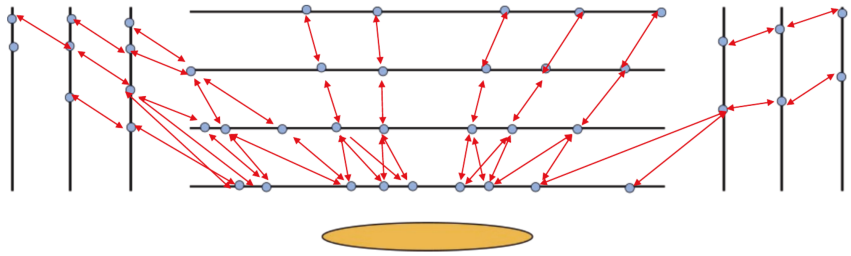


Figure 15.2: Cellular Automaton track seeding approach. First all the doublets are formed between pairs of layers, then doublets sharing a hit are connected to form triplets and quadruplets. Figure from [173].

The CA algorithm offered improvements in computational throughput, total tracking efficiency and the overall fake track reconstruction rates as detailed in [173]. Comparison between the algorithm used earlier and the CA algorithm for 2016 and 2017 detectors are shown in Figure 15.3.

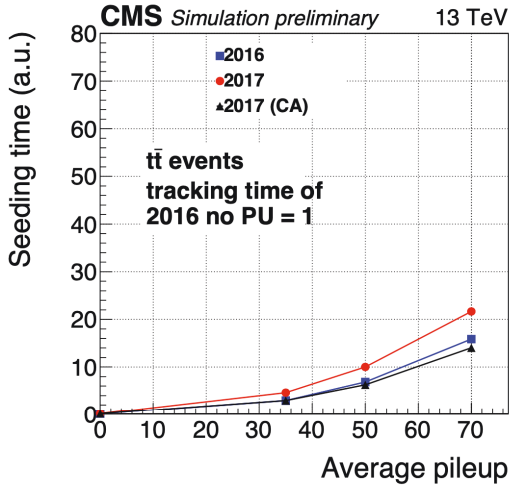


Figure 15.3: Comparison of required computing time in track seeding as a function of pile-up. Additionally the reference of 2016 reconstruction without pile-up is shown and even with the increase of charged tracks in the detector, the redesigned track seeding algorithm is able to outperform the old one. Figure from [173].

The reasoning between the different categories of track seeds becomes evident in Chapter 16, where the Iterative Tracking approach is presented in detail. It allows for the reconstruction software to start by reconstructing the most well defined and easiest to reconstruct tracks

first and then mask their signals to reduce the combinatorics for the more computationally expensive track building steps to catch the difficult tracks in the detector.

15.3 Track finding

Combinatorial Track Finder (CTF) algorithm [174] is used to extrapolate the track seeds and assign more hits to them from the outer layers. It is a Kalman filter method [175–178] for track finding that is able to propagate the expected track trajectory through the detector based on initial track parameters and their associated uncertainties. The CTF algorithm behaviour can be split into four steps:

Navigation step is where the algorithm determines which of the adjacent layers can the next hit be found from. This is done based on the current parameters of the track candidate. A fast analytical propagator is used where the trajectory of the particle is approximated as a perfect helix and the particle cannot lose any energy between two layers.

Module search returns the tracker modules that are compatible with the prediction of the navigation step as candidates for containing hits belonging to the track. A module is considered compatible if the position of the intersection between the trajectory and a module surface in the layer is no more than three standard deviations of the current track candidate position uncertainty outside the module boundary. This choice is made to limit the number of considered modules as low as possible to save computing budget while preserving the efficiency of $> 99\%$ in including the correct sensors in the computation.

Hit grouping forms collections of all the hits from module groups and performs a χ^2 test to choose the compatible hits with the extrapolated trajectory, taking into account the uncertainties in the hit and the trajectory positions. Module groups are formed so that in the regions where different sensor modules of the tracker slightly overlap each other, the overlapping sensors are put into separate module groups. A track candidate can be assigned a hit from only one of the module groups per layer so that the chance of two hits occuring in one layer due to overlapping sensors is removed. Additionally a ghost hit maybe used if no compatible hits are found, to account for a hit failing to be reconstructed due to inefficiencies in the detector. This allows the propagator to include the effect of crossing the layer without updating the position information of the propagator due to the missing hit.

Parameter update adjusts the new track candidate parameters. Each original track candidate forms new track candidates by adding one of the compatible hits found in hit grouping. Then the new track candidate parameters get their trajectory parameters updated at the position of the module surface using the added hit information with the extrapolated trajectory information. For computational performance only a small number of new track candidates are retained at every step of the CTF algorithm. Best candidates to keep are selected based on χ^2 value with bonuses and penalties given for each valid hit and each ghost hit respectively. These four steps are iterated propagating the tracks outwards layer by layer and adding more hits to them. The algorithm can be terminated based on conditions like too many ghost hits being assigned to the track or when the associated uncertainties of the track parameters fall below a threshold and the track is already considered good enough. These termination

conditions are necessary to prevent wasting computation time. If the track candidate reaches a predetermined number of hits N_{rebuild}, the iterator searches for additional hits in the inner layer by taking all the found hits, excluding the hits used in the seed from them and running the propagator backwards towards the center of the detector. This can result in additional hits being found from the inner layers that can be closer to the interaction region than the original track seed or additional hits from overlapping sensors or strip layers where the seed is restricted to use matched hits.

15.4 Track fitting

Using the hit collections assigned to each track that the track finding step produced, the track is refitted with a Kalman filter and smoother. The need for this arises from how the Kalman filter propagator updates its parameters while propagating from layer to layer while working with incomplete information about the track as its based only on the hits that have already been encountered.

The Kalman filter is initialized at the innermost hit which can be in a layer closer to the beam spot than any of the hits in the track seed were. The trajectory estimate is produced by fitting the Kalman filter with the hits in the innermost layers on the track. The filter is then updated iteratively through the list of hits propagating from inner parts of the detector towards the edge. After this a smoothing filter that is initialized with the values of the Kalman filter from the outward propagation is used to propagate the track to the opposite direction, from the detector edge towards the beam spot. The track parameters at any hit of the track are then obtained from the weighted average of the parameters of these two filters. This improves the result since one of the filters has the hit parameter estimation made using the information before the sensor surface and the other has the estimation using the information from after the sensor surface.

Instead of the fast analytical propagator the fitting step uses the Runge-Kutta numerical propagator to extrapolate the trajectory from one hit to the next one. This does not make the assumptions on perfectly helical track as would be the case in homogeneous magnetic field nor does it ignore the possibility for the particle to lose energy during its propagation in the detector volume. This improves the performance especially in the $|\eta| > 1$ regions where the magnetic field inhomogeneities are the largest.

After the track fitting is completed the hit collection of the track can be cleaned from outliers that have been incorrectly assigned to the tracks. These can be detected by comparing the fitted track position and the hit positions in the same layer. If residuals between the track and hit positions exceed a threshold, the hit can be removed from the collection and the filtering and smoothing applied again on this reduced hit collection to improve the track parameter estimate. The process is repeated until none of the track-hit residuals exceed the set threshold for outliers.

15.5 Track selection

The track finding step described earlier produces a non-negligible number of tracks that do not correspond to an actual charged particle in the collision event. Such tracks are called fake tracks and their definition is based on a certain fraction of the hits in the track belonging to another track completely. The flexibility of the track finding algorithm combined with the high signal rate in the tracker resulting in multiple suitable hits when iterating over the layers in track finding can lead to these fake tracks. The fake rate i.e. the fraction of the reconstructed tracks that are fake can be reduced by applying quality requirements to the fitted tracks before accepting them into the final track collection that is produced as an end result of the track reconstruction algorithms.

During Run 1 the quality requirements used for track selection were cuts applied to the variables associated with the fitted tracks. The requirements were:

- A minimum number of layers N_{layers} with a hit assigned to the track.
- A minimum number of layers with a 3-D hit assigned to the track.
- A maximum number of layers intercepted with the track but having no hits (a ghost hit) assigned to the track
- $\chi^2/N_{\text{dof}} < \alpha_0 N_{\text{layers}}$.
- $|d_0^{\text{BS}}|/\sigma_{d_0}(p_{\text{T}}) < (\alpha_1 N_{\text{layers}})^{\beta}$.
- $|z_0^{\text{PV}}|/\sigma_{z_0}(p_{\text{T}}, \eta) < (\alpha_2 N_{\text{layers}})^{\beta}$.
- $|d_0^{\text{BS}}|/\delta d_0 < (\alpha_3 N_{\text{layers}})^{\beta}$.
- $|z_0^{\text{PV}}|/\delta z_0 < (\alpha_4 N_{\text{layers}})^{\beta}$.

where α_i and β are constants, d_0^{BS} and z_0^{PV} are known as the track's impact parameters and defined as the distance from the beam spot center in the transverse plane with respect to the beam and the distance along the beam from the closest pixel vertex respectively. The impact parameter uncertainties δd_0 and δz_0 are computed from the fitted track trajectory. Another set of uncertainties $\sigma_{d_0}(p_{\text{T}})$ and $\sigma_{z_0}(p_{\text{T}}, \eta)$ for the impact parameters was used, where the uncertainties were parametrized based on the p_{T} and polar angle of the track.

The tighter the cuts assigned on the track quality the smaller the fake rate would become. However this is a trade-off between the efficiency in reconstructing the true tracks and rejecting the fake tracks. By tuning the constants α_i and β. CMS track reconstruction provides three working points named loose, tight and high-purity offering increasingly strict track selection requirements so that the end user could choose the best working point for their use case depending on if the analysis would benefit more from smaller fake rate or better track reconstruction efficiency.

For Run 2 the track selection algorithm moved from one dimensional cut based selections to using a machine learning based Boosted Decision Tree (BDT) algorithm to classify the tracks instead. Using simulated tracks where the knowledge which tracks are true and fake is available these decision trees were trained to use a collection of track parameters from the

track fitting listed in Table 15.1 to determine to which category the track belongs to and output its prediction as a floating point number between -1.0 and 1.0 corresponding to fake and true respectively. As some true tracks look less convincing than others and some fake tracks might have many true track like qualities in their variables, the outputs would form a distribution such as the one depicted in Figure 17.1. Based on some predetermined fake rate thresholds a working point could now be chosen to form the loose, tight and high-purity working points.

Variable	Description
p_{T}	Transverse momentum of the track
N_{dof}	Number of degrees of freedom used in track fit
N_{layers}	Number of layers with a measured hit
$N_{\text{3-D layers}}$	Number of layers with a measured 3-D hit
$N_{\text{lost layers}}$	Number of layers with a ghost hit
$[\chi^2/N_{\mathrm{dof}}]_{\text{1D mod.}}$	Normalized goodness-of-fit with a correction for 1D hits
χ^2/N_{dof}	Normalized goodness-of-fit
$\lvert\eta\rvert$	Track η absolute value
$p_{\text{T, unc.}}/p_{\mathrm{T}}$	Relative p_{T} uncertainty
$N_{\text{valid hits}}$	Number of valid hits in track
$\min(N_{\text{lost inner}}, N_{\text{lost outer}})$	Min. of # inner and outer layers invalid hits
f_{lost}	Fraction of invalid hits over all hits
$\lvert d_0\rvert$	Transverse distance from beam spot
$\lvert d_z\rvert$	Longitudinal distance from beam spot
$\lvert d_0\rvert_{\mathrm{PV}}$	Transverse distance from primary vertex
$\lvert d_z\rvert_{\mathrm{PV}}$	Longitudinal distance from primary vertex

Table 15.1: The BDT input variables. As can be seen from the variables, the BDT requires feature engineering by hand so some of the variables are given in base and modified forms to the algorithm. Variables include both kinematic track variables and variables describing the hit pattern used, in addition to impact parameters of the beam spot and primary vertex with respect to the track.

The training procedure was done using simulated events of some physics process that would produce a variety of different types of tracks encountered in the LHC collisions such as a top quark pair production event with pile-up. The BDTs give a lot more flexibility in the selection of regions in phase space that get classified as fake or as true tracks as it is able to learn these regions based on the statistical distributions of the events shown in training. The BDT classifiers will be discussed more in Section 17.1, where they are compared with the novel deep learning based approaches studied for Run 3.

15.6 Summary

The five steps presented above give a modular approach to track reconstruction. Initially all signals in the detector passing some predetermined noise level are reconstructed as hits

signalling a passage of a charged particle through the sensor during the collision event, and subsequently tracks are built by first organizing compatible hits into track seeds yielding initial kinematic parameters of the particle, which can in turn be propagated outwards in the detector using a Kalman filter to find additional hits that make up the track. After all suitable hits are found, the track is built using two Kalman filters propagating to opposite directions and combining their outputs to gain the best possible estimates of the track parameters. This approach is used for most of the tracks in the CMS detector and it produces the main track collection. Some cases like the tracks made by electrons require additional refinements on the base algorithm to account for the significant energy loss due to bremsstrahlung that distorts the trajectory. The adjustments needed to the algorithm to account for these effects in track finding are presented in [179, 180] and the Gaussian Sum Filter algorithm that is used to perform the final fit on tracks whose energy loss distribution is non-Gaussian can be found from [181].

Using different conditions on the track seeding step, focus can be put on tracks resulting from different processes in different parts of the detector. Using triplets or quadruplets of hits formed in the pixel detector layers one gets a high quality prediction on the initial track parameters that is likely to result in an accurate propagation of the track during the track building phase where additional hits are found. However many tracks might be displaced from the primary vertex due to being generated from a secondary vertex resulting from a decay of an intermediate particle in the reaction or interactions between the particles and the detector material. Also some tracks might not generate a suitable pixel seed due to sensor inefficiencies. In order to catch these tracks, additional iterations of the process starting from the seed building but using different conditions in selecting the seeding layers and the tracking region can be run. This method of iteratively running the CTF leads to higher overall efficiencies without prohibitively large computational costs if the tracks that are easiest to find are built in the first few iterations and the hits associated with accepted tracks are masked from the following iterations to reduce the combinatorial complexity of the problem. In CMS event reconstruction this multiple pass approach is called iterative tracking.

Chapter 16

Iterative Tracking

Various physical processes can cause significantly differing signals in the tracker. Some particles created in the collision can live long enough to be significantly displaced from the interaction region before decaying. This can lead to well defined track signals that are not compatible with coming from the primary vertex. Also nuclear interactions with the detector material can cause significant deflections or additional particles in being created in the charged particle trajectory. Some beyond the Standard Model theories predict exotic charged particles that can decay into or be created from a decay of a particle that is not visible to the tracker, leading to highly energetic but very short tracks created in the tracker volume.

16.1 Iterations

In order to be able to also reconstruct the various track types, the track reconstruction algorithm can be run with track seeds generated using different tracking regions and seeding layers. This allows increased efficiency in the track reconstruction procedure, since tracks that are not seeded in the algorithm will not get reconstructed. However loosening the track seeding constraints causes significant increases in the combinatorics of the problem as the number of seeds that need to be extrapolated into tracks increase. Also with looser constraints the fake rate i.e. fraction of hit combinations that are not caused by a charged particle that get used as track seeds increases.

The CMS experiment solves this problem by performing multiple passes of the track reconstruction algorithm, more specifically iterating over steps from 2 to 5 presented in Chapter 15 multiple times by using different tracking region and seeding layers for the track seeding step. The different seed categories was already presented earlier, but Table 16.1 collects the various iterations used during Run 2, the track seeds used in them and a high level description of which types of tracks are targeted with the iteration.

In total Run 2 uses ten iterations for track reconstruction. Additional two iterations are performed after the regular tracks are built for finding muon tracks, but they are excluded from the discussion here as it is more related to muons than tracking in general. The general philosophy is to try and reduce the number of unassigned hits in the tracker quickly as the

Iteration	Track seed	Description
Initial	Quadruplet pixel seed	Prompt tracks with $p_{\mathrm{T}} > 0.6$ GeV
LowPtQuad	Quadruplet pixel seed	Prompt tracks with $p_{\mathrm{T}} > 0.15$ GeV
HighPtTriplet	Triplet pixel seed	Prompt tracks with $p_{\mathrm{T}} > 0.55$ GeV
LowPtTriplet	Triplet pixel seed	Prompt tracks with $p_{\mathrm{T}} > 0.2$ GeV
DetachedQuad	Quadruplet pixel seed	Displaced tracks
DetachedTriplet	Triplet pixel seed	Displaced tracks
PixelPair	Pair pixel seed	Recovers prompt tracks with a missing hit
MixedTriplet	Triplet mixed seed	Recovers displaced tracks with a missing hit
PixelLess	Triplet strip seed	Very displaced tracks without pixel hits
TobTec	Triplet/pair strip seed	Very displaced tracks without pixel hits
JetCoreRegional	Pair pixel/mixed seed	Prompt tracks in dense jet environments

Table 16.1: Different tracking iterations used during Run 2. The iterations are executed from top row towards the bottom of the table, with the goal of significantly reducing the number of unassigned hits left in the tracker before reaching the iterations with very weak constraints on the track seeds. The first few steps are responsible for most of the reconstructed tracks, while the rest are either aiming to reconstruct displaced tracks from nuclear interactions or recovering tracks that were possibly missed by the earlier iterations due to detector inefficiencies.

iterations go along. Using hit quadruplets in the first few iterations naturally restricts the number of track seeds as the quadruplets are constrained with compatibility requirements. They also offer a good measurement of the initial parameters of the tracks, leading to a large fraction of correctly reconstructed high quality tracks as an output. Once the initial iterations have assigned a large portion of the hits in the detector, the seeding constraints are loosened step-by-step to increase the overall efficiency by finding also the tracks that are missing some of the hits due to detector inefficiencies or have been produced by some nuclear interaction making the track seeds to be too far from the interaction region or even completely outside the pixel detector for them to be reconstructed in the pixel seeded iterations.

The iterations shown in Table 16.1 have the following tasks: **Initial** iteration produces the bulk of the prompt tracks using high quality quadruplet seeds to reduce the combinatorics and reduce the fake rate in the produced track collection. It is followed by the **LowPtQuad** step where the track seeds with lower p_{T} are constructed from the pixel quadruplets. Figure 16.1 shows how this enhances the efficiency especially at the $p_{\mathrm{T}} < 0.5$ GeV region. Once the quadruplet steps aiming at prompt tracks have cleaned the hit collection, the **HighPtTriplet** and **LowPtTriplet** essentially repeat the process but by relaxing the track seeding to use triplets. They can be be seen to improve the efficiencies in the same regions as the preceeding steps. **DetachedQuad** and **DetachedTriplet** iterations relax the pixel seed constraints of promptness, allowing for the tracks that are displaced from the primary vertex to be reconstructed. These are tracks such as the ones resulting from b-quark decays. Figure 16.1 left side shows the location of these displaced tracks as a function of radial distance to the primary vertex. **PixelPair** and **MixedTriplet** steps recover tracks that are missing one hit in the pixel detector and are not being included in the pixel triplets for that reason. They

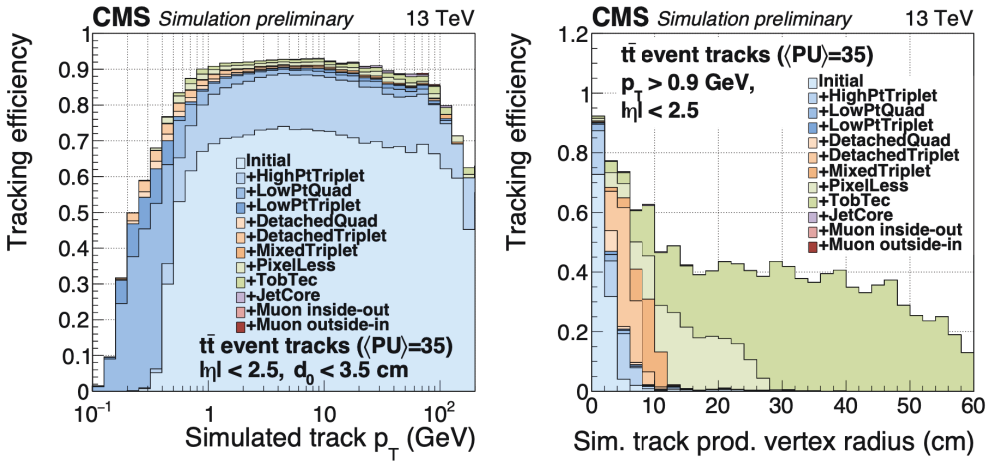


Figure 16.1: Tracking efficiency of different iterations with respect to the track p_{T} (right) and radial distance of the production vertex from the nominal beam spot (left). The shown plots are stacked histograms, so the upper limit of the histogram columns is unity. The contributions of different iterations to different regions of the track phase space is evident here. Especially the necessity of the TobTec iteration is highlighted, since its contribution to the overall efficiency is not necessarily huge but its still responsible for being able to reconstruct any tracks that are displaced by more than 25 cm from the beam spot. Figure from [173].

target the prompt and displaced tracks respectively. **PixelLess** and **TobTec** iterations are targeting tracks that are completely lacking pixel hits due to being extremely displaced from the primary vertex. These tracks can be for example a result of a nuclear interaction that a particle undergoes inside the tracker volume. They are seeded using triplets and pairs of strip seeds and these iterations are prone to producing large numbers of fake tracks since the track seeding conditions are very relaxed. However their contribution is vital in increasing the tracking efficiency for tracks created outside the innermost regions of the tracker as can be seen in Figure 16.1. The **JetCoreRegional** step has a very specific task of regaining tracking efficiency in the dense jet core regions where even the high granularity pixel detector might not be able to separate two hits from different tracks as they are too close to each other. This step allows some of the hits to be split if they look like they are a result of two hits merging together. The iteration selects the track seeds from pixel pairs only in regions where a jet has been found. While the overall increase in efficiency with this last iteration seems small, its necessity is emphasised when running the PF algorithm presented in 14.1 since the assignment of energy deposits in the calorimeters depends on whether there are a charged tracks leading to the cluster or not.

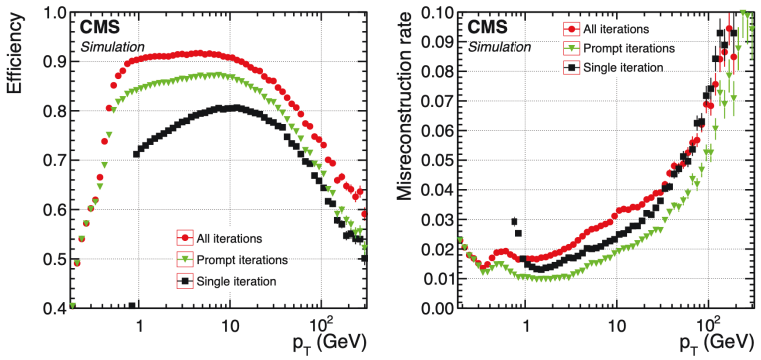


Figure 16.2: Tracking efficiencies for all, prompt and **single** iterations. Here prompt iterations contain the iterations that require at least one hit in the pixel detector to be present in the track seed, where as all iterations include the iterations targeting the displaced tracks without any pixel hits. The efficiencies are for tracks with $|\eta| < 2.5$, $d_0 \leq 3.5$ cm and $d_z \leq 30$ cm produced in multijet events without pile-up. Figure from [162].

16.2 Performance

The success of this approach is shown in Figure 16.2. Efficiencies for tracks in the parts of the detector covered by the tracker ($|\eta| < 2.5$) and originating within 3.5 cm from the beam axis and within ±30 cm from the nominal center of the CMS along the beam axis are displayed for all, prompt and a global (single) iteration. Tracks are from simulated multijet events without pile-up. Prompt refers to the iterations where at least a single pixel hit is required in the track seed. The misreconstruction rate increases when all iterations are used but this is the side effect of trying to also build the displaced tracks, as is seen when comparing the prompt and all iterations misreconstruction rates. These tracks require the track seed constraints to be relaxed significantly, allowing for more hit pairs and triplets that are not compatible with being caused by the same particle. However the increase in efficiency and ability to reconstruct tracks resulting from secondary interactions is worth the increase in fake tracks. The significant increase of misreconstructed tracks at high p_{T} values and the decrease in efficiency is partially due to the tracks being likely found inside energetic jets where multiple hits might get merged into one in the tracker. Finding all the tracks in these environments is hard, and rejecting the fake tracks using the track parameters becomes also more difficult. However the PF algorithm mitigates this effect by combining the tracker information with the calorimeter and muon system signals.

Due to the immediate performance gains in improving overall efficiency in finding and reconstructing tracks and the computational advantage that the iterative approach was roughly twice as fast as the global approach, iterative tracking quickly became the default approach in CMS track reconstruction.

Chapter 17

Classifying reconstructed tracks

At the end of each iteration of the tracking algorithm the fitted track candidates are evaluated for quality of the reconstruction. The goal of this evaluation is to separate the correctly reconstructed tracks from the misreconstructed ones – the true tracks from the fake tracks. Originally the CMS collaboration used simple cuts on a handful of the reconstructed variables to reject the most obvious misreconstructions. This method was upgraded to a machine learning based BDT classifier that used a larger set of engineered features to perform the classification. Since each iteration starts of with significantly different track seeds, a dedicated BDT was trained for each iteration to account for this fact. This method was used during the Run 2 data taking. During this thesis work implementing this track classification with novel DNN based algorithms was researched. The salient features in this new approach are reducing the reliance on explicit feature engineering, taking advantage of the capacity of the neural network to be able to use a single classifying network for the different iterations instead of training dedicated networks for each one and provide overall improved performance in track classification. Not only does the high performance in track classification translate to high quality data, it can also translate to a computational speed-up, when the true tracks get accepted with a higher rate and their associated hits get masked from considerations of the following iterations.

17.1 Boosted Decision Tree classifiers

As was mentioned in Section 17.1 the BDT algorithm is trained to perform subsequent cuts on the input variables. Based on the used hyperparameters the algorithm then runs over the training data set multiple times producing N_{trees} decision trees. Each run over the dataset is used to train a new decision tree in a manner that minimizes the loss function being optimized. This additive model improvement is referred to as *boosting* in the name of the algorithm. As an end result an ensemble of decision trees with N_{trees} individual classifiers is produced and the output of the BDT classifier is the average over all the trees. In order to avoid overtraining the model, each individual decision tree is usually limited to using only a few of the available variables to make a couple of selection cuts. Although this leads to any single decision tree to give poor performance on the task, the collective of the decision trees

is able to learn to classify the inputs.

Since each step in the iterative track reconstruction produces a set of tracks with possibly significantly differing distributions in some of the track parameters. For example the impact parameters of the prompt iterations are by necessity constrained so that the tracks have been created within the pixel detector while the very displaced tracking iterations use track seeds in the strip detector without much requirements on the impact parameters. For this reason each iteration gets their own classifier trained specifically on track reconstructed by that iteration. Since the classifier's decision is used to mask the hits in accepted tracks from the following iterations to reduce the combinatorics, the process cannot be parallelized so that one could train the classifiers of each iteration at the same time. Instead the training has to proceed one iteration at a time and new training data has to be reproduced so that all the newly trained classifiers of the preceeding iterations are in place before. This aspect makes training or retraining the BDT classifiers a tedious and slow task, leading to rigidity in the work flow if for example the detector conditions change in the tracker in a way that has an effect on the variables used for the classification. The cumbersomeness of the BDT training in this context is one of the motivations in moving onto more flexible algorithms like the DNNs that could learn all the iterations in one go. The approach in selecting the samples for training the classifier is to try and include as many of the different track types encountered during deployment. Usually a sample with $t\bar{t}$ events and pile-up or multijet events with pile-up are used since there the tracks targeted by different iterations of the algorithm are present. However there may be occasional need to enrich the training sample with specific track types to improve the performance, for example by including more high p_{T} jets from other samples to train the classifier of JetCoreRegional iteration or additional electron tracks from $Z \to e^+e^-$ events since they have a slightly different distribution of track variables due to the bremsstrahlung phenomenon.

The track reconstruction at the CMS experiment relies on the implementation of the Boosted Decision Tree algorithm provided in the TMVA 4 [182, 183] package of ROOT [184]. It offers a framework for training and testing many different multivariate analysis methods including Boosted Decision Trees. There are many variants of the algorithm and below is only the description of the algorithm in the form that it has been used for track classification in the CMS experiment. The necessary hyperparemeters and training samples for performing the training are discussed below.

The important hyperparameters in the BDT training are collected and explained in Table 17.1. N_{trees} determines the number of individual trees that are trained to make up the algorithm. In general a large number of trees indicates better behaviour of the algorithm as the trees are grown to complement each other using the boosting algorithm. However the more trees there are the longer the algorithm takes to train. d_{depth} limits how many variables the single tree uses to make a yes or no decisions to classify the sample. Larger depth can offer more fine grained classification of the samples, but it can also lead to overtraining the algorithm if the number of samples ending in the leaf nodes starts to get too small to represent the whole distribution. This means the tree might end up learning just a statistical fluctuation in the training dataset. N_{cuts} controls how many threshold values of the variable the algorithm evaluates before deciding what is the optimal cut value. Increasing the number

of values to evaluate can result in improved classification, but it increases the computational cost of training the algorithm. *BaggingFraction* determines how large fraction of the training set is used for training any single tree in the collection. This introduces a stochastic element to the training as different trees see slightly different examples during training. *Shrinkage* controls how much weight any single tree is given. Essentially this reduces the learning rate of the model as any single tree gets their contribution reduced by the shrinkage factor. This can increase the amount of trees required for good performance, but it is generally considered to improve the performance of the model. *BoostType* determines which boosting algorithm to use in the training. Gradient boosting algorithm used here is considered good with decision trees. In this implementation a binomial log-likelihood loss is used to optimize the classification:

$$L(F, y) = \ln\left(1 + e^{-2F(x)y}\right), \tag{17.1}$$

where $F(x)$ is the model response to input x and y is the target value the model should learn to assign to this input. The larger a discrepancy between the model response and the target value the higher the loss value as can be seen when considering the target values in the binary classification problem are -1 and 1 for the two classes, and the model predictions are bound to the interval $[-1, 1]$. Since the model outputs a continuous value within the interval as its prediction, how close it is to either target value can be considered as a confidence of sorts. The model is maximally wrong when it outputs a value of -1 for a sample in class 1 or vice versa. As the model response is the result of all M trees in the collection voting on what the output value should be, the response function can be considered as a weighted sum of parametrised base functions$f(x; a_m)$ that are often referred to as "weak-learners" – i.e. the individual decision trees in the collection. The model response can be then written as

$$F(x; P) = \sum_{m=0}^{M} \beta_m f(x; a_m); P \in \{\beta_m; a_m\}_0^M. \tag{17.2}$$

In order for the collective behaviour of the model and as such the model response $F(x)$ to evolve in the direction that minimizes this loss function, each new tree in the collection is grown by setting the tree parameters to match leaf values to the mean value of the gradient in each region. That way every new tree moves the predictions in the regions that it's structure considers towards a direction that minimizes the loss function based on the value of the gradient. This way even though any single weak-learner would do a poor job in predicting the classes, the additive behaviour of all the weak-learners in the model manages to output a good prediction since adding all the gradient steps together leads the model response to be near the (local) minimum of the loss function once training has converged.

Parameter	Description	Used value
N_{trees}	Defines the number individual trees to be trained	2000
d_{max}	Maximum depth of a tree i.e. how many selections does a tree make	3
N_{cuts}	Number of grid points in variable range used to finding the optimal node splitting cut	20
BaggingFraction	Uses a randomized subset of the full dataset of this size for each tree	0.5
Shrinkage	Weight of an individual tree	0.1
BoostType	Boosting algorithm to use	Grad
Training datasets	The simulated events used in training the classifiers	TTbar with PU 35 High p_{T} QCD (for JetCoreRegional)

Table 17.1: The training parameters needed for the BDT training using TMVA framework.

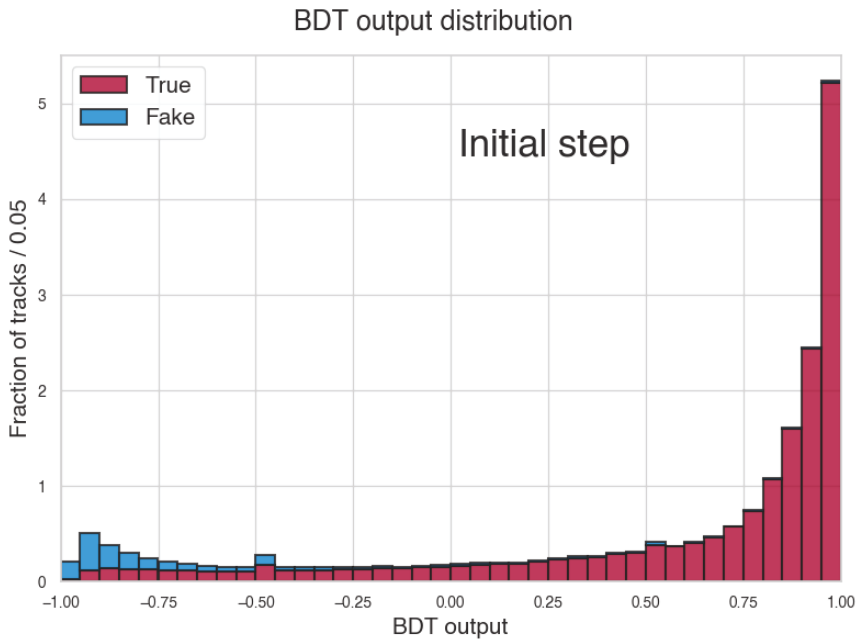


Figure 17.1: Output distribution of the BDT classifier in the initial iteration for a multijet sample with pile-up in Run 2 conditions. The bin height shows the fraction of reconstructed tracks in the iteration that get assigned the the bin value by the BDT. In the initial step of the reconstruction, the quadruplet seed takes care that the ratio of misreconstructed tracks stays relatively small compared to the correctly reconstructed tracks. By requiring the BDT output value to be above some threshold, the sample can be further purified from the fake tracks.

An example of the BDT algorithms output values for a sample of multijet events with pile-up in Run 2 conditions is shown in Figure 17.1. The binning is normalized so that the bin heights multiplied by the bin widths summed together equals unity. The distribution shows that the BDT which has been trained to classify the true (training target of +1) and fake tracks (training target of -1) in the event provides a good estimate if the track is correctly reconstructed or misreconstructed. It is important to note that even at the very lowest of output values near -1.0, there are some true tracks present. This simply means that what looks like a true or a fake track is not unambigous, since fake tracks can produce the same track parameters as a true track would. Hence the best the classification algorithms can do, is to learn statistical trends in the distributions of true and fake tracks and try to best optimize its outputs so that there is maximal separation between the outputs for the two classes. Based on the output of the classifier, the user can decide on how much they appreciate having maximal efficiency in reconstructing the tracks versus how much they need to remove the fake tracks from their analysis.

The BDT algorithms are often appreciated due to being easy to use and often having an excellent performance "out-of-the-box", with little tuning of the training parameters required. Additionally the gradient boosting algorithm allows the use of robust loss functions that are not sensitive to outliers and restricts the individual decision trees to be weak-learners protecting the classifier from overtraining. This makes it a sensible algorithm as a first go in solving a problem and they have provided excellent track classification performance in the CMS detector track reconstruction.

Downsides of the BDT algorithm include the requirement to explicitly feature engineer the input variables, which can require significant intuition in the application domain and the problem at hand. The necessity of this becomes obvious when considering that the predictions made by the decision trees are just a result of sequential one dimensional selection cuts on the input variables, so the algorithm only produces partitions the feature space into hypercubes and gives an output value for each hypercube. This does not allow for combining the input features into new, possibly better combinations that could improve the performance of the algorithm. This means that the algorithm cannot always attain the theoretical best performance since it is restricted to the input variables as given. However in real life use cases this is rarely a problem and the BDT produces excellent performance when compared to many other classifier algorithms.

For the track classification the motivation to move to other algorithms for the classification stem from possibility of improved performance in cleaning the misreconstructed tracks out of the samples. This becomes increasingly relevant as the luminosity of the LHC increases, producing more pile-up collisions in each collision event. The additional signals in the tracker from the pile-up tracks increase the number of fake tracks being reconstructed since there is a limit on how much the constraints on the track seeds can be tightened without losing tracking efficiency. The track classification algorithm is the last line of defence that can prevent the increase of fake rate with the increasingly challenging detector conditions. The BDT algorithm from Run 2 will be used as a baseline reference for the studies presented next.

17.2 Deep Neural Network classifier

Deep neural networks have become a natural next step to study in search for improvements over the BDT algorithm. They have shown excellent performance in a wide variety of domains due to their extreme flexibility as was described in Chapter 9.

17.2.1 Feature engineering

Deep neural networks have shown impressive performance in a wide variety of tasks in physics and other fields. They have become also a natural next step to explore for additional performance over the BDT algorithm. Most appealing feature in the DNNs is their ability to implicitly learn new representations of the input variables by combining the information of the different inputs in the hidden layers of the model. This reduces the reliance on the explicit form of the input variables given to the network. Additionally DNNs excel at compacting

high dimensional input spaces into something computationally manageable so that the number of input variables to use can be extremely large if necessary. This often leads to DNNs being able to perform well with an input that consists of a large number of low-level features that the network can then learn to combine into higher level representations as necessary during training [185]. In [186] this was demonstrated in a context more familiar to particle physics, where a search for an exotic particle was performed using a BDT with high-level feature engineered input variables often used by experimental physicists in these searches and its performance was compared to DNNs trained with either the same high-level features, high-level features and additional low-level features, and only with the low-level features. Not only were the DNN approaches achieving improvements on the classification of the signal, the network trained using only the low-level features was able to achieve the same level of performance as the network trained with high- and low-level features. So the networks are able to extract at least as much information as is contained in feature-engineered inputs from the low-level set of inputs that can be combined to produce the high-level features. However this does not mean that the networks cannot take advantage of high-level features by for example converging more quickly to a good solution, it just demonstrates it is not strictly necessary at least in this type of use case.

17.2.2 Capacity

Another aspect of the DNNs that is taken advantage of has to do with the capacity of the model to learn a large number of patterns in its training. It was shown in [187] that a shallow two layer DNN can be trained to learn unstructured random noise perfectly as long as the number of parameters exceeds the number of data points. This is a unsurprising consequence of the universal approximation theorem for neural networks, stating that with a finite number of neurons and one hidden layer, the network can approximate any continuous function arbitrarily well [188].

While this is a dire warning that overfitting the neural network is an issue that has to be considered, it also allows one to consider simplifying the process of training classifiers for the iterative track reconstruction algorithm of the CMS experiment. It is certainly within the guarantees of a universal approximator to be able to learn to classify true and fake tracks reconstructed in ten different iterations using different track seeds instead of having to train ten separate networks to do the task. Additionally the ten different networks could be considered as a single network branching into ten branches after the first layer that switches on the branch relevant for the iteration used to reconstruct the sample. With this in mind the task was set to study how a single classifier network could be produced that could handle the task. The benefits on top of speeding up the (re)training process by removing the sequential portion of it, it would make the supporting code for deploying the network into the CMS reconstruction software cleaner. Also since only the weights of a single classifier would have to be kept in runtime memory when executing the code, there is room for improved memory footprint as long as the size of the neural network does not grow too large.

17.2.3 Variable preprocessing

The BDTs are able to perform fine without preprocessing the input variables in any particular way since the one dimensional decisions are made based on the particular variables own scale where different selection thresholds are estimated. Neural network learning dynamics are more sensitive to the scale and distribution of the input variable values. One way to quickly verify this is by looking at the activation functions used in DNNs, and noting that many of them might have their "sensitive" region somewhere between [-1, 1] surrounded by a saturated region where the value of the derivative of the activation function goes to zero.This is not strictly the case for every activation function as most notably the ReLU activation does not conform to this, and the neurons can learn a bias term that corrects the inputs closer to the sensitive region. One can even argue that the neural network can learn to perform what ever scaling it needs in the first few layers should it be advantageous. However having the inputs scaled to lie around zero is shown to help improve stability, convergence speed and performance of neural networks. The concept has been around for a while [189] and it can already be considered an industry standard. Especially when working with algorithms like Batch Normalization [190] in the network training where the input is expected to come in with approximately mean of zero and unit variance.

Another aspect of neural network input preprocessing has to do with how to present categorical variables. When presenting data to a neuron, the different values of the input variable are imposed to have some metric between them due to the activation function that is applied. More concretely when giving as inputs 1.0 and 2.0 to a neuron with some activation function, the neuron might give a different pair of outputs than for the input pair of 1.0 and 5.0. For strictly monotonic activation functions the output values of the first pair will be closer together than the output values of the second pair. If the inputs are really describing some value where a concept of distance is meaningful, like the p_{T} of the particle, this is not a problem. Indeed it could be considered a desirable attribute. But if the values 1.0, 2.0 and 5.0 refer to the classes "cat", "dolphin" and "dog" respectively having them treated as if 1.5 would signify a class that is halfway between a cat and a dolphin would make little sense. Such categorical variables are better treated differently. A common method is to one-hot encode the variables so that based on the number of possible classes n, the single integer input describing the class label is converted into a vector of n elements.

The approach taken here includes also preprocessing the input variables to the network. The list of used variables is presented in Table 17.2. The motivation behind the selection of the variables has been a result of trial-and-error through starting from the variables used before in the BDT implementation and adding more variables to the mix. Due to the computational demand of the offline track reconstruction, some restrictions on what types of variables can be used has to be taken since some variables might take significantly longer to compute. Variables stored to the reconstructed track are good candidates as they do not need to be recomputed.

Out of the used variables the ones where the relative sign of the input is not considered important the absolute value is taken, e.g. track η since the important information is contained in knowing how forward was the track not in which half of the detector it was. The variables that contain a large and possibly sparsely populated scale, a transformation of

Variable	Description
p_{T}	Transverse momentum
$p_{\mathrm{x,\ inner}}$	Momentum in x-coordinate at the innermost hit
$p_{\mathrm{y,\ inner}}$	Momentum in y-coordinate at the innermost hit
$p_{\mathrm{z,\ inner}}$	Momentum in z-coordinate at the innermost hit
$p_{\mathrm{T,\ inner}}$	Transverse momentum at the innermost hit
$p_{\mathrm{x,\ outer}}$	Momentum in x-coordinate at the outermost hit
$p_{\mathrm{y,\ outer}}$	Momentum in y-coordinate at the outermost hit
$p_{\mathrm{z,\ outer}}$	Momentum in z-coordinate at the outermost hit
$p_{\mathrm{T,\ outer}}$	Transverse momentum at the outermost hit
$\mathrm{Err}_{p_{\mathrm{T}}}$	Uncertainty in p_{T}
$\lvert d_0\rvert_{\mathrm{PV}}$	Transverse distance from primary vertex
$\lvert d_z\rvert_{\mathrm{PV}}$	Longitudinal distance from primary vertex
$\lvert d_0\rvert$	Transverse distance from beam spot
$\lvert d_z\rvert$	Longitudinal distance from beam spot
$\mathrm{Err}_{\lvert d_0\rvert}$	Error transverse distance from beam spot
$\mathrm{Err}_{\lvert d_z\rvert}$	Error longitudinal distance from beam spot
χ^2/N_{dof}	Normalized goodness-of-fit
η	Track η
ϕ	Track ϕ
Err_{η}	Uncertainty on track η
Err_{ϕ}	Uncertainty on track ϕ
N_{pixel}	Number of pixel layers with a measured hit
N_{strip}	Number of strip layers with a measured hit
N_{dof}	Number of degrees of freedom used in track fit
$N_{\mathrm{lost\ inner}}$	Number of lost hits in inner layers
$N_{\mathrm{lost\ outer}}$	Number of lost hits in outer layers
$N_{\mathrm{inactive\ inner}}$	Number of inactive sensors crossed in inner layers
$N_{\mathrm{outer\ inner}}$	Number of inactive sensors crossed in outer layers
$N_{\mathrm{lost\ layers}}$	Number of layers with invalid hits
OrigAlgo	Label for the iteration that first reconstructed the track

Table 17.2: The DNN input variables. During preprocessing, for the first 17 variables in the list (green) first the absolute value of the variable is taken and then the natural logarithm of $1 + x$ is computed where x is the variable. For η (red) just the absolute value is taken. OrigAlgo (blue) is one-hot encoded as described in Section 17.2.3. Then all the variables except OrigAlgo are scaled to the interval $[-1, 1]$ using the minimum and maximum values of the variables in the training set. During deployment all input values are clipped based on this minimum and maximum value that get stored into the model so that the model will only see values within the range it saw during training.

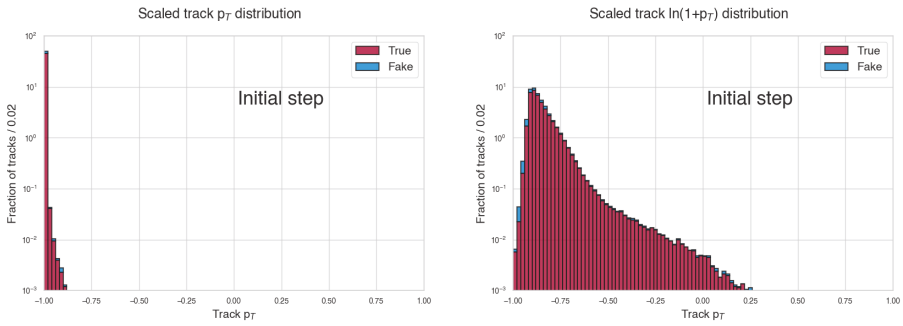


Figure 17.2: Track p_{T} distribution scaled to the input range of $[-1.0, 1.0]$ without taking the natural logarithm (left) and with the natural logarithm transformation (right). The used tracks are ones built in the Initial iteration of reconstructing multijet events with pile-up in Run2 conditions. While the transformation does not bring any inherently new information to the network, having the distribution more spread out in the interval provides better performance during training and deployment.

$x \to \ln(1 + |x|)$ is performed. This enhances the network classification performance both in training and deployment. This is possibly because of preventing the values from becoming extremely squeezed during rescaling to the input interval of $[-1, 1]$ due to large outliers in some distributions. This is shown for the track p_{T} distribution in a multijet sample with pile-up in Figure 17.2, the input distribution to the network is shown both when the natural logarithm transformation is done and when it is not.

While these transformations do not bring any additional information for the network, and in fact when taking the absolute value some information is discarded, this leads to more stable network performance. Especially when inspecting the classifier performance with respect to variables like the p_{T} of the track where the training distribution falls exponentially towards higher values, this type of transformation that improves the denseness of the input samples in the variable range seems to help with avoiding strong correlations of the classifier output and the variable. This helps with the common issue where the classifier ends up being very confident in classifying the tracks at higher p_{T} values either as true or false seemingly at random. This type of failure modes for the outlier values in the training distributions are a known issue [191, 192] in many real world applications and the phenomenon is referred to as the long-tail recognition problem. Although majority of the research in the field revolves around recognizing rare classes in multi-class identification problems, the methods are applicable to the binary problem faced in track classification as well. More specifically the different iterations each generate true and fake tracks but with different distributions for the track variables used as inputs leading to within-class imbalances of different "subconcepts" i.e. iterations. In order to be able to use only a single network as a classifier for all iterations, this imbalance has to be accounted for when selecting the training samples.

17.2.4 Training data

Similarly as with training the BDT algorithm a general sample containing most of the track types encountered in proton-proton collisions is desirable as a starting point for the training. Suitable candidates are QCD multijet events or $t\bar{t}$ production events with pile-up tracks. This approach worked well with the BDTs, with the only required adjustment being that the JetCoreRegional iteration of the track reconstruction algorithm had to be trained with a special sample of QCD jets containing higher p_{T} jets. While these objects are present in the regular multijet events as well, the probability to generate a jet of given p_{T} falls exponentially as a function of p_{T}. This leads to a shortage of the energetic tracks within the jet center regions in the regular training samples and as a consequence produces a classifier that underperforms on these tracks.

The neural networks are capable of producing more finely grained predictions using the same input variables as the BDT due to not being restricted to hypercubes formed by the one dimensional selection cuts. This leads to a situation analogous to overfitting the network, where the performance when deployed to the real world is significantly lower than expected if the training samples are not representative of the real distribution that will be encountered. This ended up being one of the significant challenges in developing the DNN based track classifier. For example the general track samples of QCD multijet events with pile-up have strong within-class imbalances between different iterations. This is by construction as the initial iterations are meant to reconstruct a significant fraction of the tracks to reduce the combinatorics in the later iterations. The fraction of all tracks binned by the iteration that first reconstructs them is shown for QCD multijet events with pile-up in Figure 17.3, where they are also compared to the same histogram for tracks from Z$\rightarrow e^+e^-$ events. The imbalance between different iterations can lead to the classifier being undertrained with respect to some samples due to examples in e.g. DetachedQuad iteration being relatively rare during the training.

Common methodology for handling such cases is to resample or reweight the training events in such a way that it presents the samples in a more balanced way. Although giving different samples weights based on their perceived "importance" i.e. giving the rare samples higher weights has been a popular choice in the field, recent empirical results [193–195] indicate that the sample reweighting approach might not work as expected and does not produce a meaningful difference once the training has converged, and may in fact hinder the training process by slowing it. However sub-sampling the training set has a demonstrable effect on the learned function, and that approach is followed here as well.

Even with sub-sampling, additional measures are taken to ensure the network performance. Additional datasets are included in the training to ensure track types that are known to be rare in the QCD multijet events are included to enrich the training sample, similarly as the QCD high p_{T} jets were used for training the BDT for JetCoreRegional iteration. The included samples contain electron tracks ($Z \rightarrow e^+e^-$ events), high p_{T} jets (QCD multijet events containing a jet with 1800 GeV $\leq p_{\mathrm{T}} \leq$ 2400 GeV) and very detached tracks from a SUSY process where a stop quark is created in the collision and it decays into a b-quark and chargino ($\tilde{t} \rightarrow b\tilde{\chi}_1^+$) where the chargino subsequently decays into a W^+ boson and a neutrino. This leads to charged tracks that are displaced from the primary vertex.

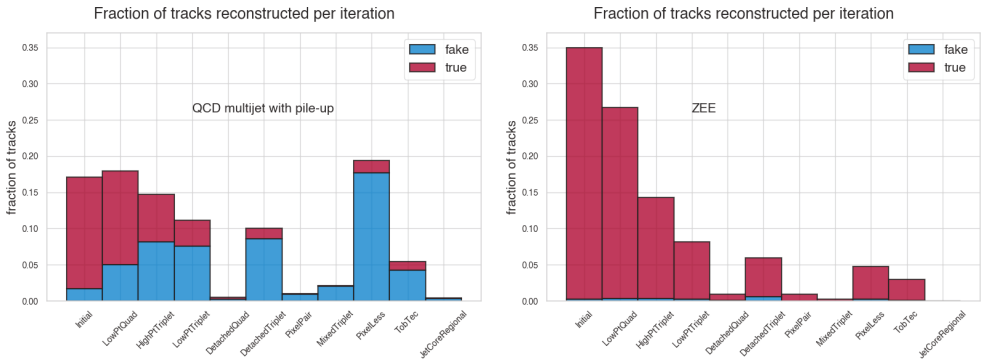


Figure 17.3: True and fake track fractions per iterations for QCD multijet events with pile-up (left) and ZEE events (right). The height of the bin shows the fraction of all tracks that are first reconstructed in that iteration. As the track classifier is expected to work well on all the various processes the CMS track reconstruction might encounter, care has to be taken that the classifier also learns to classify correctly the PixelLess tracks created by electron in the ZEE events, even though the multijet sample is dominated by fake tracks in that iteration.

In Figure 17.4 the distribution of true and false tracks to different iteration is shown after subsampling the combined training sample. In order not to throw away tracks meant for enriching the sample, the subsampling is done in two stages: First the QCD multijet sample with PU is subsampled to contain equal amounts of tracks from different iterations, then the samples from different processes are added and the dataset is subsampled again. This is due to practical concerns since the QCD multijet samples are used in many different purposes for validating the software, and as such there is a high availability of large statistics for these events. This is usually not the case for the more specialized samples. The raw QCD multijet dataset before subsampling contains O(100 million) tracks, but the final subsampled training set has roughly 10 million tracks, where 6 million are from the QCD multijet events and the rest are evenly sampled from the other datasets.

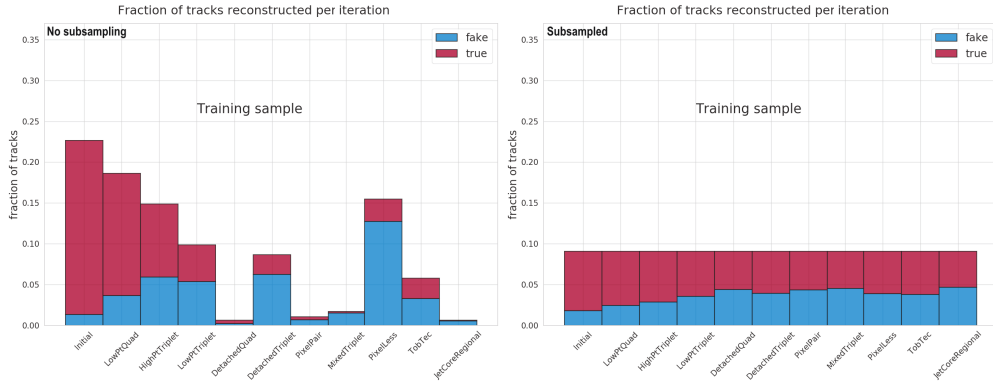


Figure 17.4: The training sample dataset before (left) and after (right) subsampling each iteration to have same level of representation in the training set. While the number of fake and true tracks are not equal in different iterations, their ratios are not as extreme as the QCD multijet sample alone would produce, due to the contributions from the additional samples producing more specialized tracks like the electron tracks or displaced tracks.

17.2.5 Hyperparameters and network architecture

Due to the nature of the variables used as inputs for the network, an approach of densely connected hidden layers with common regularization methods of l2 weight decay, batch normalization and dropout were chosen for this task. In order to check that there is no obvious problems that would end up having the network underperforming, a hyperparameter optimization using KerasTuner [196] framework was performed, where different parameters like the number of layers, neurons in a layer, activation functions, learning and dropout rates, l2 regularization weights and optimizers were tested. The optimization was performed using the Hyperband algorith [197] which speeds up the standard random search in hyperparameter optimization approach [198] through improved resource allocation and early-stopping.

The Hyperband algorithm search space is summarized in Table 17.3. This covers a variety of dense network specifications and also tests if the one-hot encoding and input preprocessing approaches described are effective by allowing the network to test turning both of those layers on or off during the hyperparameter search. *Activation* makes the choice of activation functions between rectified linear units (ReLUs) and Swish [199]. These are two common choices for dense networks that tend to work well in many different contexts. The latter was discovered by an extensive automated search for new activations functions [200] performed in order to find functions that could improve on the impressive performance of ReLU. The number of neurons in a layer starts with the first layer using N_{units} neurons and each subsequent layer using $\max(N_{\text{units}}/i^2, 32)$ where i is the running index of the hidden layers. This leads to a network shape that decreases quickly in the number of trainable parameters deeper in the hidden layers. N_{layers} controls how many hidden layers are included in the network. *Onehot* and *Preprocess* switch on or off the use of the respective input transforming layers. r_{dropout} and $r_{\text{l2 reg.}}$ control the dropout rate between layers and and L2 regularization strengths respectively. r_{learning} sets the learning rate of the optimizer. *Optimizer* selects either Adam or

Parameter	Possible values (sampling strategy)
Activation	ReLU, Swish (choice)
N_{units}	1024, 512, 256, 128, 64, 32 (choice)
r_{dropout}	$[0.0, 0.5]$, (float, uniform sampling)
N_{layers}	$[3, 7]$, (integer, uniform sampling)
$r_{\text{learning rate}}$	$[1^{-5}, 1^{-3}]$, (float, logarithmic sampling)
Onehot	True, False (choice)
Preprocess	True, False (choice)
$r_{\text{l2 reg.}}$	$[1^{-7}, 1^{-2}]$, (float, logarithmic sampling)
Optimizer	Adam, SGD (choice)

Table 17.3: Search space for the Hyperband algorithm. The possible values or ranges of values and the sampling strategy are listed for each parameter.

SGD optimizers to perform the updates. Adam is the industry workhorse used in variety of contexts, while SGD is a useful comparison due to being a simple optimizer.

The search algorithm ran 200 tests exploring the configurations of the search space and each test was allowed to run at most ten times over the training set of O(10 million) tracks. The algorithm split the models into brackets of two, where each model was trained for two epochs before the less performing model is dropped and the winner proceeds to the next bracket and is trained for two more epochs and so on until the ten training epochs are up. Out of these tests, the five best performing configurations were chosen based on achieving the smallest validation loss values at the end of training. These best performing configurations are presented in Table 17.4.

From the best models a few features stand out. Each of the models use preprocessing layer and the Adam optimizer. This is however expected based on input preprocessing being a standard procedure in the field and the Adam optimizer being specifically designed to improve on the flaws encountered with the SGD algorithm. For the other parameters the main observations would be that there is a large range of suitable values for them. I.e. the network does not seem to be too picky on the values of these hyperparameters. Perhaps surprisingly the *Onehot* parameter is not set to True for all of the networks. While the argument presented earlier about the one-hot encoding still holds, not using it does not remove the information from the input variable, it just makes it more difficult for the network to digest due to the input format enforcing a metric distance between different values which is not sensible when discussing track reconstruction iterations.

As the search algorithm only runs at maximum ten epochs of the training, the five best models are further studied to understand their performances with respect to each other. Each model is trained using five different random initialization of the weights and random shuffle orders of the training samples in order to verify that the architectures provide consistent performance. The training is also run longer, each iteration of each model being trained for 50 epochs through the training dataset in order for the training weights to have time to convergence. The results of these trainings are collected in Figure 17.5, where on the left side the validation loss values during training are presented and on the right the validation ROC AUC values

Model label	Parameters
Model 0 Trainable params: 10689	*Activation* =ReLU $N_{\text{units}} = 128$ $r_{\text{dropout}} = 0.15$ $N_{\text{layers}} = 4$ $r_{\text{learning rate}} = 1.75^{-4}$ *Onehot* =False *Preprocess* =True $r_{\text{l2 reg.}} = 5.1^{-4}$ *Optimizer* =Adam
Model 1 Trainable params: 11009	*Activation* =Swish $N_{\text{units}} = 64$ $r_{\text{dropout}} = 0.20$ $N_{\text{layers}} = 8$ $r_{\text{learning rate}} = 1.0^{-4}$ *Onehot* =False *Preprocess* =True $r_{\text{l2 reg.}} = 1.9^{-7}$ *Optimizer* =Adam
Model 2 Trainable params: 16001	*Activation* =Swish $N_{\text{units}} = 128$ $r_{\text{dropout}} = 0.25$ $N_{\text{layers}} = 6$ $r_{\text{learning rate}} = 1.42^{-4}$ *Onehot* =True *Preprocess* =True $r_{\text{l2 reg.}} = 2.2^{-4}$ *Optimizer* =Adam
Model 3 Trainable params: 108841	*Activation* =Swish $N_{\text{units}} = 512$ $r_{\text{dropout}} = 0.4$ $N_{\text{layers}} = 8$ $r_{\text{learning rate}} = 6.24^{-5}$ *Onehot* =True *Preprocess* =True $r_{\text{l2 reg.}} = 5.37^{-7}$ *Optimizer* =Adam
Model 4 Trainable params: 14785	*Activation* =ReLU $N_{\text{units}} = 64$ $r_{\text{dropout}} = 0.45$ $N_{\text{layers}} = 10$ $r_{\text{learning rate}} = 8.55^{-4}$ *Onehot* =True *Preprocess* =True $r_{\text{l2 reg.}} = 2.83^{-7}$ *Optimizer* =Adam

Table 17.4: Parameters of the five best performing model search as found by the Hyperband algorithm. The color coding matches the result plots below.

are shown. The values are calculated as mean values of the five training iterations of each model and the bands around these values show the minimum and maximum values from the five iterations.

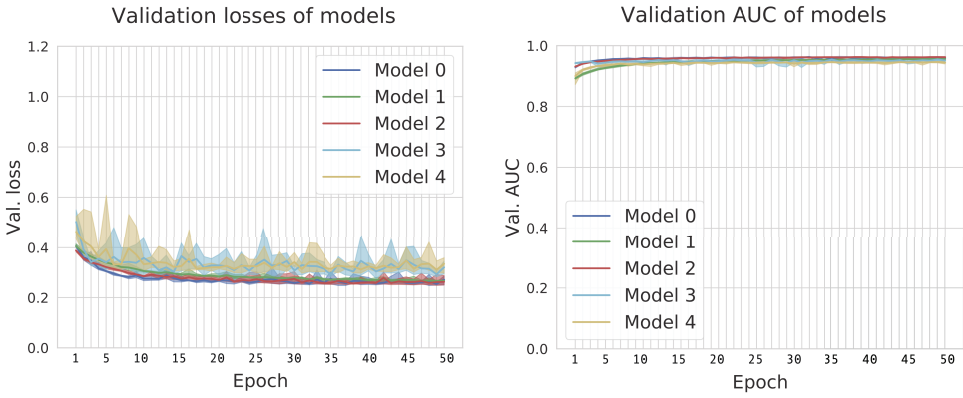


Figure 17.5: Results of retraining the five best models from the hyperparameter search. Each model is trained five different times with random reinitialization of weights and shuffling the training dataset in order to see that the models give consistently good training performance. The value at each epoch is set as the mean value of the five different iterations while the bands around it display the minimum and maximum value from the five trainings. **Left**: The validation losses during training epochs. **Right**: The validation ROC AUC values during training.

All the models reach comparable level of performance. For Model 3 and Model 4 there is large variance between the five training runs of the model. Based on their hyperparameter values the likely cause is the relatively high value on the r_{dropout} parameter combined with the large number of layers. Higher rates of dropout between the layers tend to require a longer time to converge into a stable performance, so the variance might decrease if the training were to be run longer. Model 0, Model 1 and Model 2 all reach similar level of performance on the used metrics after the initial 20 or so epochs. The variance between different initialization of the models is small.

Based on the results shown here in this task the neural network architecture seems to be fairly robust over a wide range of hyperparameters as demonstrated by the configurations of the five best performing networks studied above. The only clearly useful settings are the use of preprocessing on the non-categorical inputs and the use of Adam optimizer over SGD optimizer.

An architecture based on Model 2 is chosen for the actual training and deployment for testing. It provides good performance with a reasonable number of parameters, indicating that the runtime performance regarding computing time and memory footprint can be kept low. Schematic describing the used model is shown in Figure 17.6. The network training uses early stopping technique, where the network is allowed to train until validation loss

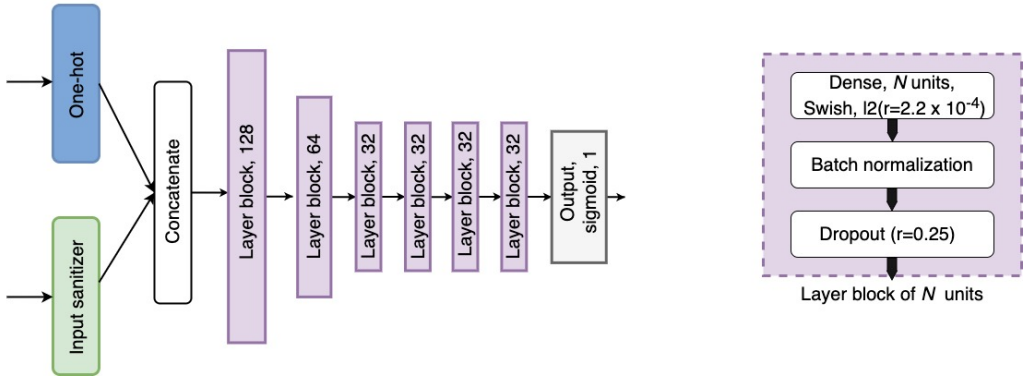


Figure 17.6: **Left**: The network architecture based on Model 2. **Right**: Each layer block contains a dense layer with Swish activation followed by batch normalization and dropout.

has stopped improving for several epochs at which point the weights are rolled back to the configuration with the lowest validation loss.

17.3 Performance

In order to gain a realistic understanding of the classifier performance, it is tested on multiple datasets. The BDT classifiers used for Run 2 are used as the baseline that the new classifier should be compared against. The datasets used for producing performance plots are independent from the datasets used in training to avoid biasing the results. In the following a tracking particle is defined as a simulated particle expected to leave a reconstructable track in the detector.

The performance plots are produced using a preselected collection of tracks satisfying $|\eta| \leq 3.0$, 0.005 GeV $\leq p_{\mathrm{T}}$, the tracking particles are charged and tracks from out of time pileup from earlier or later bunch crossings are excluded. **Efficiency** is defined as fraction of tracking particles that are associated to tracks in the detector over the number of tracking particles. **Fake rate** is the fraction of tracks not associated to any tracking particles over the number of tracks. **Duplicate rate** is the number of tracks associated to a tracking particle that has been associated to two or more tracks over the number of tracks. **Pileup rate** is the number of tracks associated with a tracking particle created by in-time pileup event.

In the plots presented the baseline is with respect to the tracking final selection being performed by the BDT classifiers. The DNN results refer to the final selection being performed with the deep neural network classifier. All the studies presented here are done using simulated tracks. The goal of retaining at least the same efficiency as the baseline has while reducing the fake rate and pile-up rate as much as possible. There are two working points inspected, *all tracks* which aims to contain as many of the tracks corresponding to actual charged particles in the collisions as possible at the cost of increasing fake rate and *high purity*

where only tracks of high quality are retained so that they can be used in analyses sensitive to the presence of fake tracks. The type of tracks can vary significantly from process to process so the results are presented and discussed with respect to multiple different datasets corresponding to different hard interactions at bunch crossing.

The plots are produced and inspected using the same workflow as is used in the CMS collaboration's tracking validation.

17.3.1 QCD multijets with pile-up dataset

The QCD multijet sample with pile-up represents perhaps the most general sample of tracks present in the detector giving a good overall performance metric for tracking. The densely populated core areas of hadronic jets provide a challenge for the jet core regional track reconstruction step and short-lived hadrons can provide displaced vertices, tracks that split into multiple new tracks due to decays taking place and tracks that are created further away from the beamspot. The inclusion of pile-up makes this sample representative of the actual conditions during the detector operations where there is a significant amount of tracks unrelated to the primary vertex that are being reconstructed.

In Figure 17.7 the efficiency and fake rate in the QCD sample with pile-up is shown with respect to reconstructed track p_{T}. The new classifier shows significant reductions in fake rate of up to 50% compared to the baseline BDT classifier in the all tracks working point. In the high purity working point there is a region around 1 GeV where the DNN classifier seems to be underperforming slightly with respect to the fake rate but there is correspondingly a slight surplus in efficiency. This means there is room to adjust the selection cut value used for the high purity working point to lower the efficiency slightly in order to reduce the fake rate.

When inspected with respect to η in Figure 17.8 the tracking efficiency displays no non-uniformities in its performance when comparing the two classifiers. In fake rate there are slight variations seen especially around the transition regions of the tracker detector. This implies a difference in how the trained classifier has learned to take advantage of this information.

Measuring the efficiency and fake rate against pile-up in Figure 17.9 the DNN classifier shows a constant improvement across the inspected pile-up range in the all tracks working point. The important aspect is that at least the classifier performance is not deteriorating as the number of vertices and as such the number of tracks in the detector increases. In an optimal case one could device a classifier that could negate the rising trend in fake rate with respect to pile-up that can be seen in both working points as this is one of the challenges faced when the luminosity of the LHC collider will increase in the future.

In Table 17.5 and Table 17.6 the numbers of tracks after the selection for both working points are collected. The notable detail is that the overall number of tracks that pass the selection and are stored in the track collection information either to be consumed by other modules in the reconstruction software or used in some other analysis decreases significantly while the number of true tracks passing the selection increases. This comparison shows that the

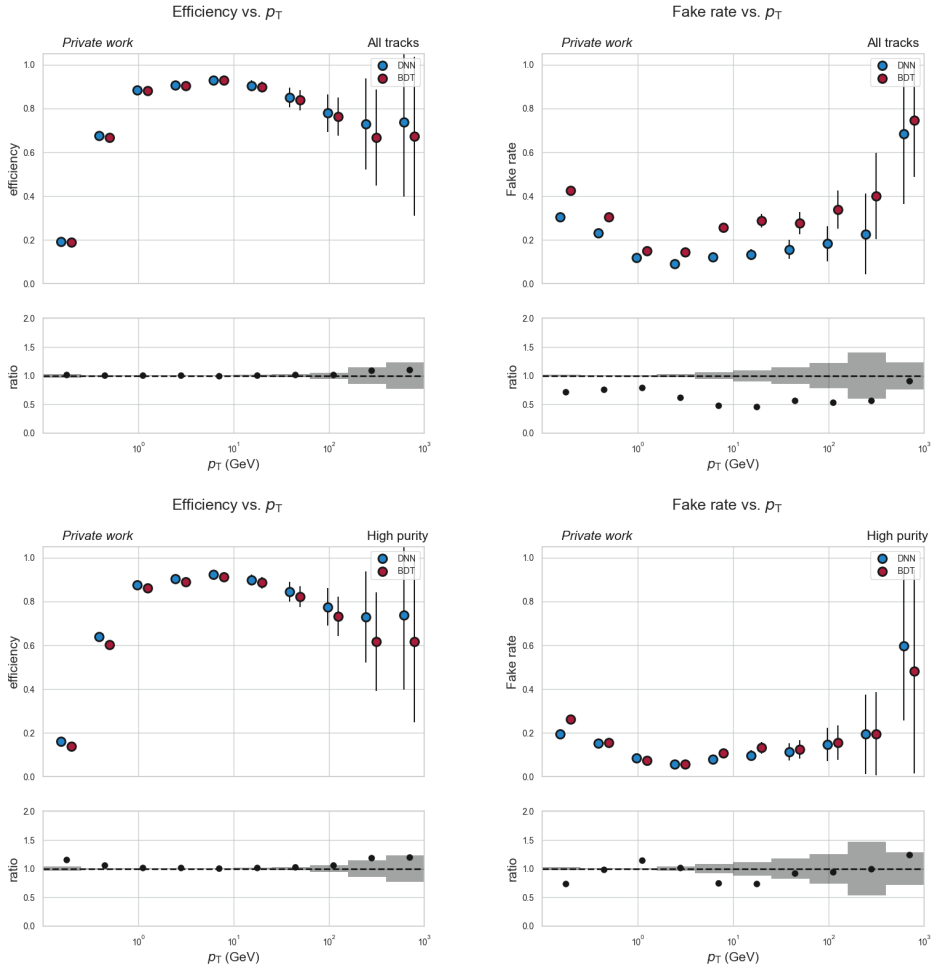


Figure 17.7: **Top**: Tracking efficiency and fake rate for all tracks against p_T. **Bottom**: Tracking efficiency and fake rate for high purity tracks against p_T. The markers are offset around the bin center for clarity. Error bars in the upper portion and grey error bands in the lower portion of the plots represent statistical uncertainty. Here the new DNN classifier is shown to significantly better remove fake tracks especially in the *all tracks* working point without reducing the overall efficiency compared to the baseline BDT classifier.

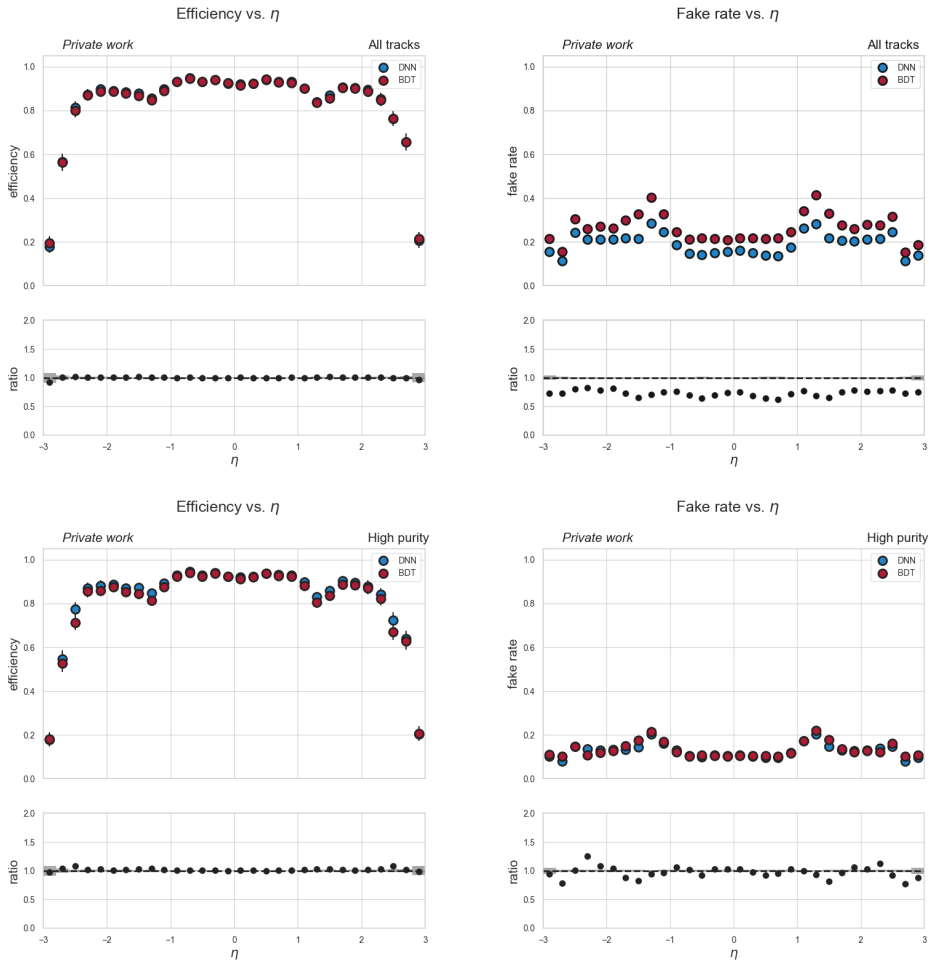


Figure 17.8: **Top**: Tracking efficiency and fake rate for all tracks with respect to η. **Bottom**: Tracking efficiency and fake rate for high purity tracks with respect to η. Error bars in the upper portion and grey error bands in the lower portion of the plots represent statistical uncertainty. Here the new DNN classifier is shown to significantly better remove fake tracks especially in the *all tracks* working point without reducing the overall efficiency compared to the baseline BDT classifier.

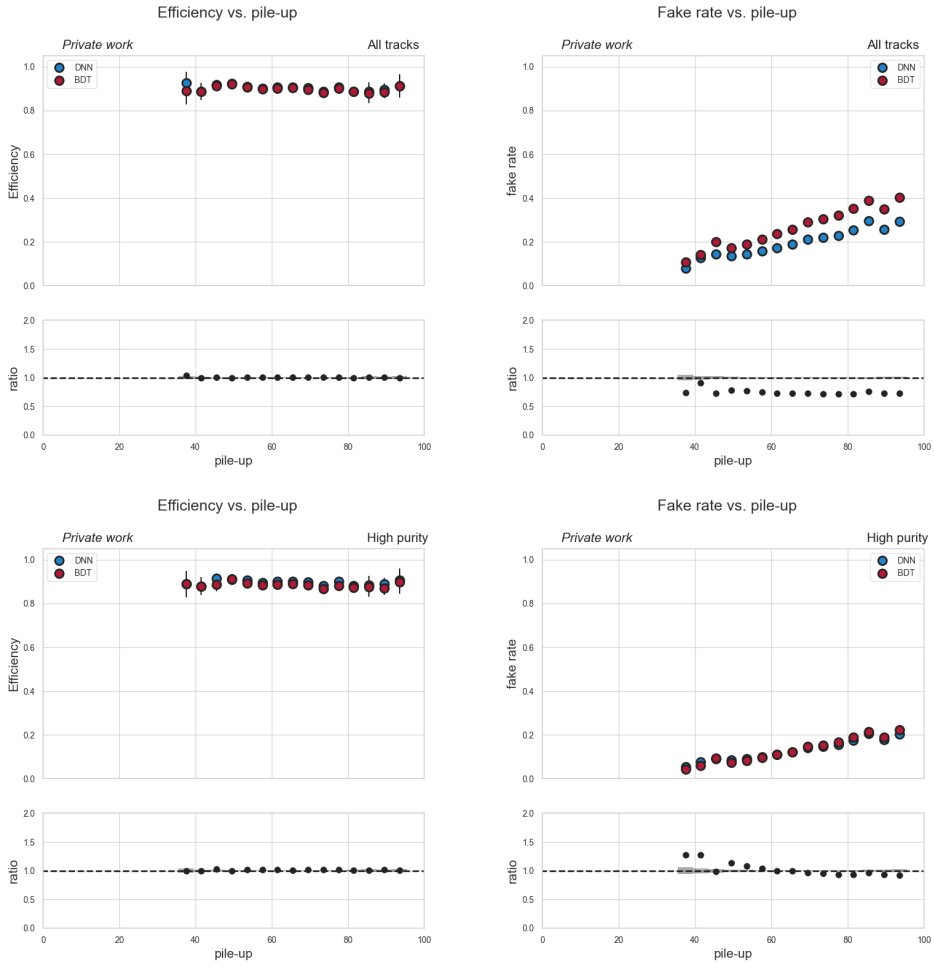


Figure 17.9: **Top**: Tracking efficiency and fake rate for all tracks with respect to the number of pile-up vertices. **Bottom**: Tracking efficiency and fake rate for high purity tracks. Error bars in the upper portion and grey error bands in the lower portion of the plots represent statistical uncertainty. Here the new DNN classifier is shown to significantly better remove fake tracks especially in the *all tracks* working point without reducing the overall efficiency compared to the baseline BDT classifier.

All tracks	BDT	DNN	Relative change
Efficiency	0.898	0.902	+0.4%
Number of tracks	1180805	1078979	−8.6%
Number of true tracks	863021	867769	+0.6%
Number of fake tracks	317784	211210	−33.5%

Table 17.5: Efficiency and numbers of different tracks passing the all tracks working point selection.

High purity tracks	BDT	DNN	Relative change
Efficiency	0.891	0.895	+0.4%
Number of tracks	242031	231937	−4.2%
Number of true tracks	205992	206767	+0.4%
Number of fake tracks	36039	25170	−30.2%

Table 17.6: Efficiency and numbers of different tracks passing the high purity tracks working point selection.

single deep neural network has significant potential in improving the track selection over the collection of boosted decision tree classifiers as it prunes away more fake tracks while passing more true tracks simultaneously.

Cutting down fake tracks in the reconstructed track collection not only removes false signals from subsequent analyses using tracking information downstream but also speeds up reconstruction algorithms that use tracking information and match them to other signals like reconstructed energy deposits in calorimeters like the particle flow algorithm. As the QCD with pile-up sample represents a very generic hard interaction taking place in proton-proton collisions, good performance of track selection algorithm in this context is imperative.

17.3.2 $Z \to e^- \bar{e}^+$ dataset

In order to specifically validate that reconstructed electron tracks are correctly classified the $Z \to e^- \bar{e}^+$ needs to be monitored for changes due to the new classifier. This dataset contains a large number of reconstructed electron tracks with a wide p_{T} range from the decay of Z boson. This was one of the issues encountered early in the development of the DNN classifier even though there are certainly electron tracks present in the more general datasets like the QCD multijet dataset. It could be that these tracks are such a small fraction that the classifier is incentivized to just classify them fake or not really learn then at all since their effect on the overall score is too small. By enriching the training sample with electron tracks this issue is remedied.

Compared to the QCD multijet with pile-up sample here one must note the fact that the absolute fake rates are significantly smaller. This is because the simulated Z$\to e^+e^-$ sample here does not have pile-up included since that would obfuscate the efficiency measurement on electron tracks. In Figure 17.10 the efficiency is shown to be retained when compared to the BDT classifiers across the p_{T} range. Importantly there is no sudden drops in performance at

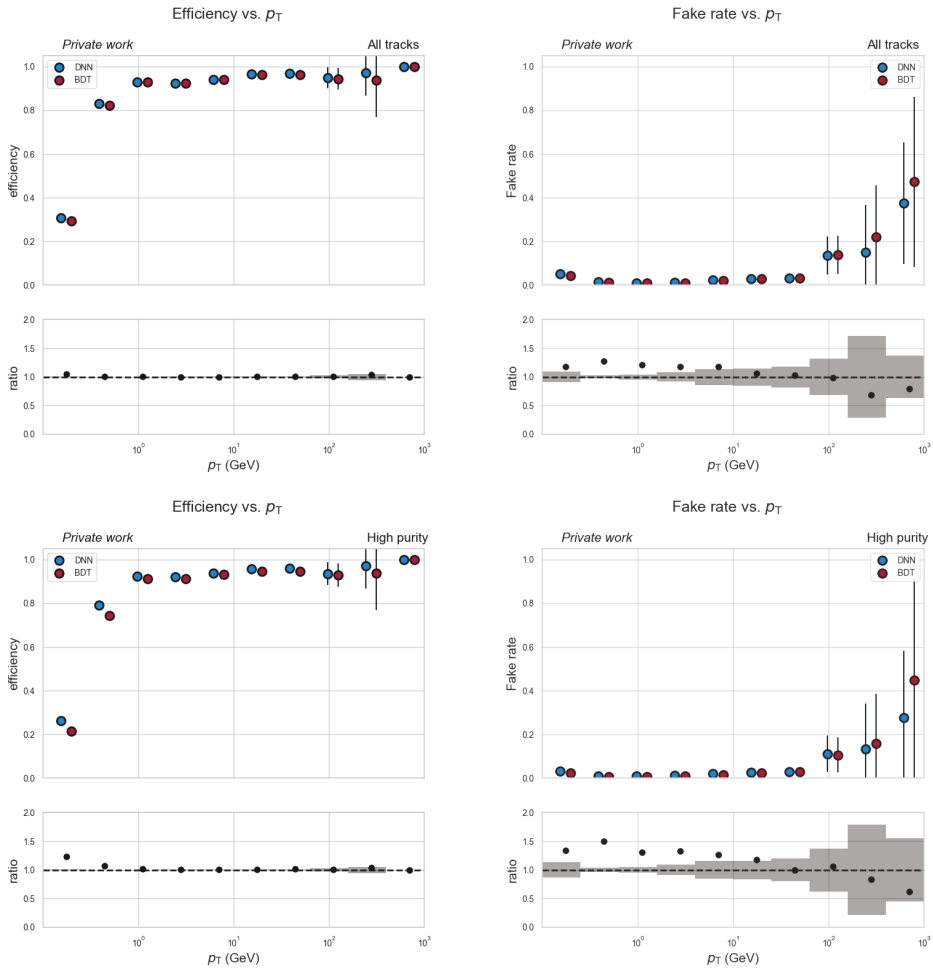


Figure 17.10: **Top**: Tracking efficiency and fake rate for all tracks against p_T. **Bottom**: Tracking efficiency and fake rate for high purity tracks against p_T. The markers are offset around the bin center for clarity. Error bars in the upper portion and grey error bands in the lower portion of the plots represent statistical uncertainty.

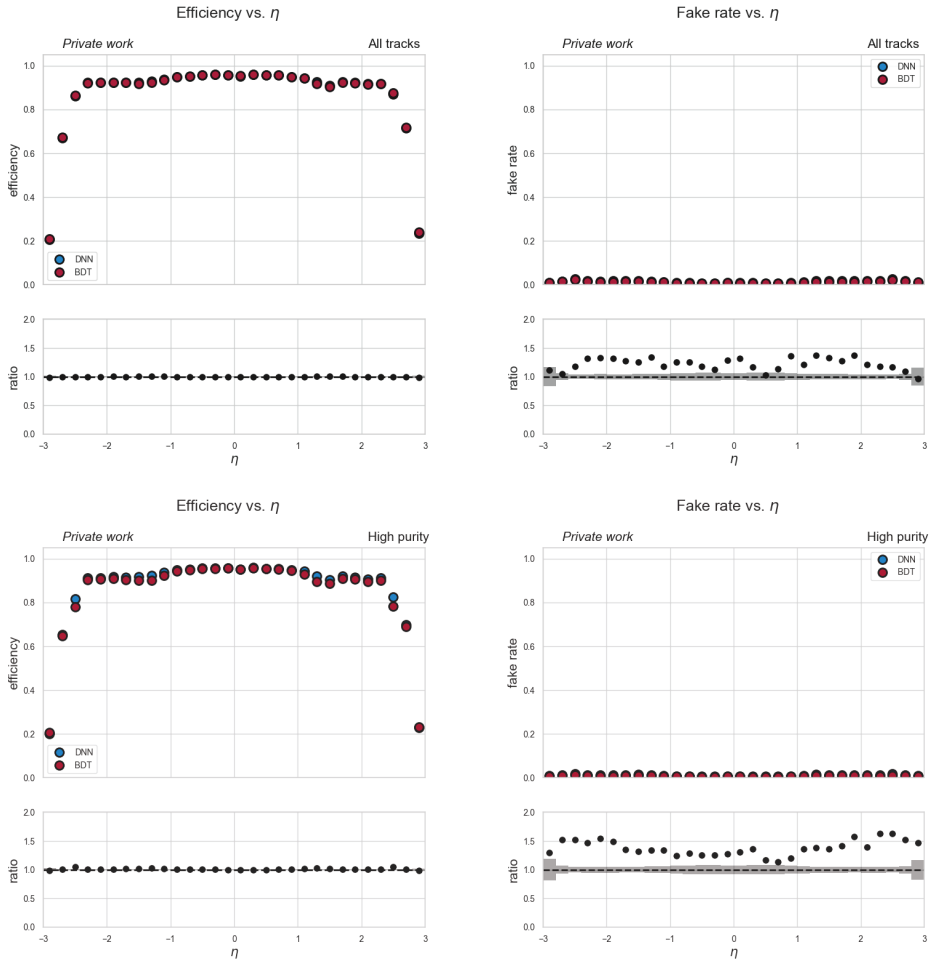


Figure 17.11: **Top**: Tracking efficiency and fake rate for all tracks against p_{T}. **Bottom**: Tracking efficiency and fake rate for high purity tracks against p_{T}. Error bars in the upper portion and grey error bands in the lower portion of the plots represent statistical uncertainty.

All tracks	BDT	DNN	Relative change
Efficiency	0.932	0.934	+0.2%
Number of tracks	750655	762754	1.6%
Number of true tracks	741868	751720	−1.3%
Number of fake tracks	8787	11034	+25.6%

Table 17.7: Efficiency and numbers of different tracks passing the all tracks working point selection.

High purity tracks	BDT	DNN	Relative change
Efficiency	0.919	0.920	+0.1%
Number of tracks	213392	214807	+0.7%
Number of true tracks	211095	212122	+0.5%
Number of fake tracks	2297	2685	+16.9%

Table 17.8: Efficiency and numbers of different tracks passing the high purity tracks working point selection.

larger p_{T} values that could be seen early on when the training sample was made up exclusively out of QCD with pile-up dataset. In fake rate the relative changes seem worrisome at first but due to the absolute fake rate being so insignificantly small the change is not of relevance.

Tables 17.7 and 17.8 show the absolute number of tracks as well as how they are split between true and fake tracks after the track reconstruction and selection is done. Although the number of fake tracks unfortunately grows for both working points, the absolute number of added fake tracks due to the change of classifier stays small even when the relative change is large.

17.3.3 SUSY

The prevalent issue during the development of the DNN classifier was catastrophic performance with some samples containing so-called exotic tracks that maybe caused for example by an electrically neutral supersymmetric particle that is created in the hard interaction and travels for some distance before decaying into a charged particle that leaves signals in the tracker layers. This would cause displaced tracks that could have a significant amount of momenta associated to them.

Partially the issue was in availability of such samples as they were originally private productions of specific groups searching for SUSY signals and not centrally produced making the usual validation sequence blind to changes there. Additionally the poor availability of the samples prevented enriching the dataset with examples of these tracks in these SUSY scenarios.

Especially in the cases where a short and energetic track is displaced in the detector, the signal looks in many aspects similar to a fake track that could arise from associating unrelated hits with each other and as such the classifier may easily end up learning to classify all such

tracks as fake. An effort to remedy this issue was made using special weights increasing the importance of the few true tracks with matching characteristics present in the general samples like QCD multijet dataset as well as other more direct approaches like bounding the input values into some predetermined ranges with poor results.

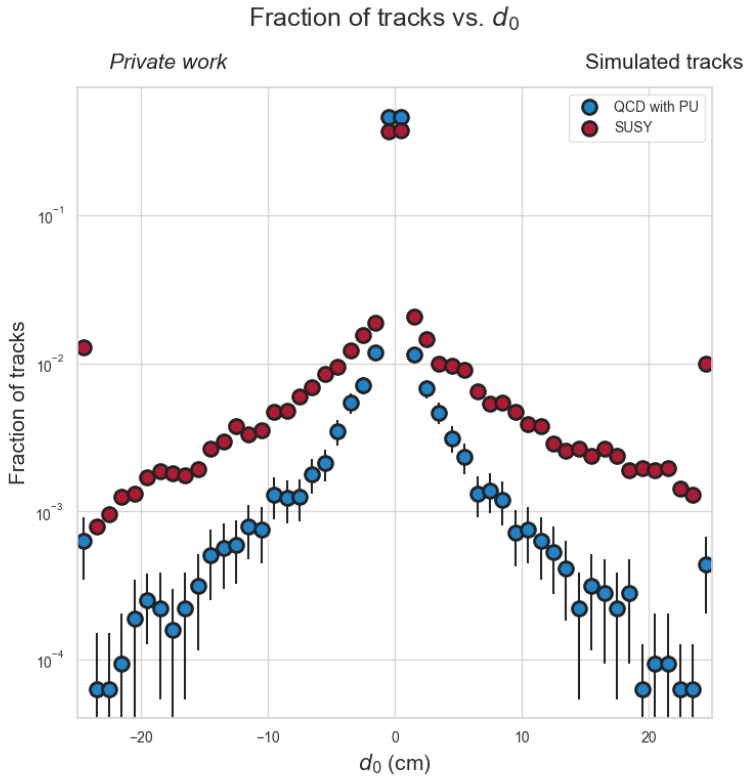


Figure 17.12: Fraction of simulated tracks with respect to displacement d_0 from the beam spot in the transverse plane. Comparison between QCD multijets with pile-up and SUSY datasets. Overflow and underflow bins are shown at the edges of the binning range.

The problem is visualized in Figure 17.12. Here the fraction of simulated tracks at some transverse distance d_0 from the beam spot is shown between the QCD with PU and SUSY datasets. While the QCD with PU dataset clearly contains tracks at high displacements from the beamspot they make up a very small fraction of the dataset compared to the distribution in the SUSY dataset. This risks the classifier to become undertrained in classifying displaced tracks of this type if the training set is composed exclusively from the QCD sample. The displacement is used here as the most obvious difference between the datasets that can be easily visualized in a histogram over a single variable. The truth is likely more complicated and in addition to the differences in single variable distributions the correlations between different variables and their combinations differ as well.

All tracks	BDT	DNN	Relative change
Efficiency	0.913	0.916	+0.3%
Number of tracks	189634	193754	+2.2%
Number of true tracks	186371	188913	+1.4%
Number of fake tracks	3263	4841	+48.4%

Table 17.9: Efficiency and numbers of different tracks passing the all tracks working point selection.

High purity tracks	BDT	DNN	Relative change
Efficiency	0.902	0.904	+0.2%
Number of tracks	77227	78188	+1.2%
Number of true tracks	75747	76259	+0.7%
Number of fake tracks	1480	1929	+30.3%

Table 17.10: Efficiency and numbers of different tracks passing the high purity tracks working point selection.

Inspecting Figures 17.13 and 17.14 show that with the training set enriched with displaced tracks the performance concerns raising from the lack of such tracks in the QCD multijets with pile-up dataset are not relevant. The efficiency of the BDT classifier is at least matched across the whole range in p_{T} and η. Similar excess in fake tracks being let through the selection as with the $\mathrm{Z}\rightarrow e^{+}e^{-}$ dataset can be seen here but in this case as well the absolute numbers of fake tracks reconstructed in the no pile-up conditions are so small that this is of not too much concern.

The absolute numbers of tracks in the collections presented in Tables 17.9 and 17.10 confirm the efficiency does not fall behind the BDT classifier as well as the relative difference in fake tracks being not as significant in terms of absolute tracks. The number of accepted true tracks is slightly above the target of matching the BDT performance so there could be room to optimize the working point further to slightly reduce the fake rate if such was considered necessary.

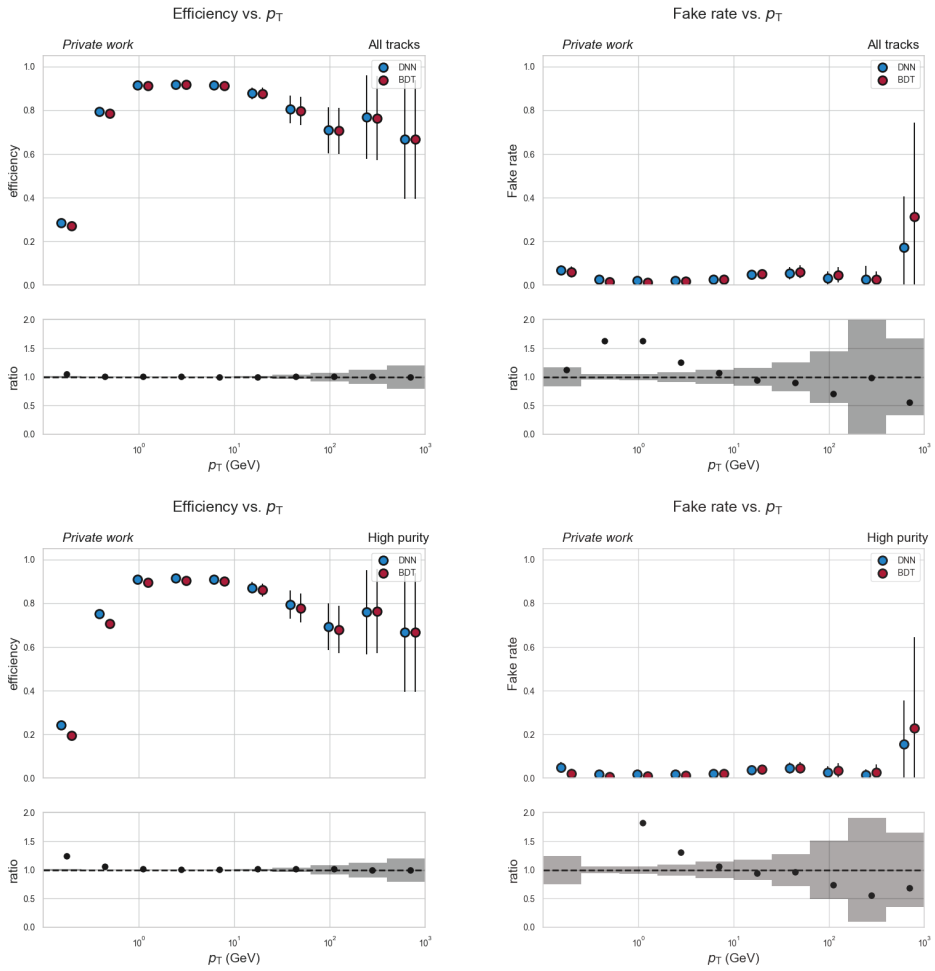


Figure 17.13: **Top**: Tracking efficiency and fake rate for all tracks against p_T. **Bottom**: Tracking efficiency and fake rate for high purity tracks against p_T. The markers are offset around the bin center for clarity. Error bars in the upper portion and grey error bands in the lower portion of the plots represent statistical uncertainty.

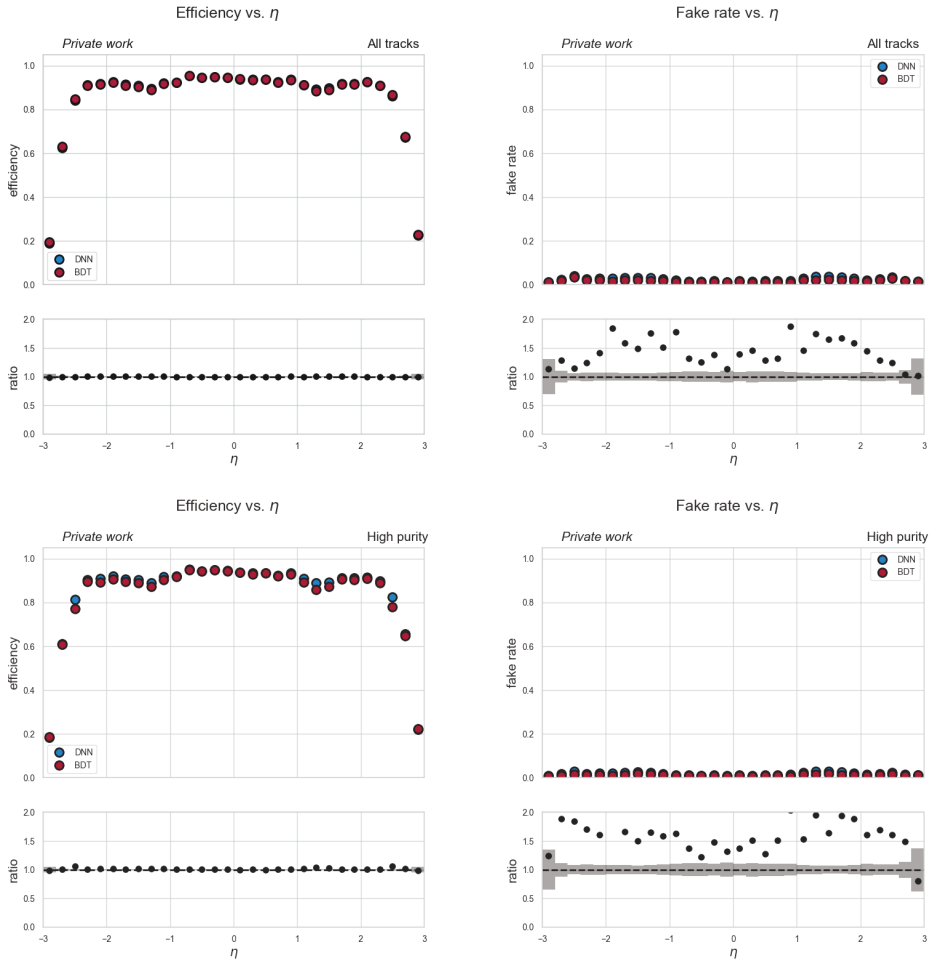


Figure 17.14: **Top**: Tracking efficiency and fake rate for all tracks against p_{T}. **Bottom**: Tracking efficiency and fake rate for high purity tracks against p_{T}. Error bars in the upper portion and grey error bands in the lower portion of the plots represent statistical uncertainty.

17.3.4 Timing and memory footprint

While good performance is necessary for the classifier algorithm the memory footprint and run time of the chosen algorithm have to be considered as well. Even though the offline reconstruction is not as time sensitive a task as for example the reconstructions needed for online trigger decisions during data taking, the computing budget for reconstructing the collected data is limited. As highlighted already track reconstruction is an expensive part of the reconstruction chain due to the combinatoric nature of the task where signals from different layers of the tracker need to be assigned to their correct tracks.

The chosen classifier algorithm has a two fold effect: On one hand a classifier that can reduce the number of fake tracks accepted in the reconstruction will speed up the downstream reconstruction chain as they will not be considered in other algorithms using the track collections. On the other hand the cost of evaluating a track with the classifier will depend on the algorithm used.

It's difficult to set definite goal posts for the memory and timing performance of the classifier. Here the inspection is only done on a very high level inspecting the fractions of memory and compute time taken up by the track classifier class during running the reconstruction chain of the QCD with pile-up events. Table 17.11 contains the relative fractions for the BDT and DNN classifiers.

The DNN classifier seems to take up a significantly larger fraction of computing time as well taking up almost an order of magnitude larger fraction of memory. While this is certainly something to take note of and investigate further to understand the cause especially for such a significant increase in memory requirements and see if it could be mitigated, even such changes are not necessary a complete disaster for the new classifier. Although the fractions spent in classification shift the overall effect is not as direct, since the overall performance of the track reconstruction algorithm will also be affected by at which tracking iteration the true tracks are accepted and what is the ratio of true tracks and fake tracks in the final track collection as well as its overall size. As was shown in Subsection 17.3.1 the amount of accepted true tracks in the all tracks working point increases while the overall number of tracks is significantly reduced as the fake rate is cut drastically. While this may come at the cost of the algorithm taking up some additional resources during runtime it can cause meaningful reduction in further computing downstream as well as the required storage for the reconstructed datasets.

	BDT (% total)	DNN (% total)
Memory	0.38	3.18
Time	0.21	0.87

Table 17.11: The relative memory footprint and computation time taken by the two classifier implementations. The percentages are with respect to the track reconstruction step.

17.4 Future work and prospects

In improving the classifier performance the next step could be to look into lower level variables available in the CMS track reconstruction chain. The approach presented above used a rather limited set of variables describing the reconstructed track that are known to be sensitive to mistakes made in the reconstruction. However there is plenty more information that could be useful for to be used as inputs such as the hit level variables including information such as the charge distributions used for reconstructing the hits in the track which should display some correlations if they have been caused by the same charged particles. Another aspect is the neural network architecture. The network presented here is in many respects the naivest possible implementation where as elsewhere convolutional neural networks and graph networks have shown great performance and could be applicable to the tracking problem as well.

As was discussed above it turned out that the original training dataset had to be augmented with various other datasets in order to account for the different track types that were not well represented in the QCD with pile-up dataset. This gives cause to consider if it is irresponsible to use a track classifier that is not exposed to the different simulated datasets that are going to be reconstructed with the tracking algorithms later on. While the training and classification performance with the BDT collection trained on a single simulated dataset did not suffer from many of the issues that could be seen during the development of the DNN classifier this could be a result of the BDT training algorithm enforcing the resulting classifier making too coarse decisions. As described in Section 17.1 each individual weak learner in the collection of decision trees is only allowed to make up to three selection cuts on variables chosen by random from the set of input variables to make the classification decision. Taking a collection of thousands of weak learners makes it probable that all the input variables can be used in the decision, but still there is no classifier that has optimized it's prediction specifically to the regions requiring the granularity of four consecutive selection cuts, for example a decision about tracks in the transition region of the tracker ($|\eta| \geq 1.2$ and $|\eta| \leq 1.6$), with high p_{T} ($p_{\mathrm{T}} \geq 10$ GeV) and large displacement ($d_0 \geq 10$ cm). Such regions are instead classified by weak learners trained to optimize a larger region of the phase space. However the DNN can and likely will use all of the input variables in its decisions and it could be the reason for the seemingly better granularity in comparison leading to possible issues if some subset of track types has not been encountered during training.

The direct solution for the issue could be to improve the training set to include a larger variety of different physics processes. As the CMSSW validation workflow already requires a sizeable production of release validation datasets it, a solution that could automate sourcing this already produced data for the training workflow to produce an extensive training dataset. However this would prevent validating the classifier performance with the release validation data as it has been used in the training, biasing the classifier. A dedicated track production just for machine learning purposes could avoid this but adds additional stress to the restricted computing resources.

17.5 Summary

The work presented above represents the very first venture in performing reconstructed track quality estimation and classification using deep neural networks in the CMS collaboration conducted during 2016-2020 by the author. During this time a huge leaps were done in deep learning across disciplines as well as frameworks for training and deployment, best practices and understanding of the abilities of these algorithms. Additionally the CMSSW framework is adjusting to more demand for deep neural networks to be deployed in multiple segments of data taking and processing. Current framework using TensorFlow interface within CMSSW provides a good ground for further development and monitoring of neural network algorithms flexibly.

The DNN implementation was shown to achieve the set goal of at least parity in efficiency compared to the BDT classifier on three key datasets while reducing the fake rate in a general setting of QCD with pile-up events. In no pile-up detector conditions the fake rate was seen to increase slightly for the Z$\rightarrow$ e^+e^- and SUSY samples. This could indicate that possible gains could be achieved by for example increasing the fraction of tracks built in no pile-up conditions during training. The runtime and memory footprint of the new classifier compared to the old one were inspected and while the DNN looks like to be a more demanding algorithm with both regards the observations made are not a deal breaker for moving onto deep neural networks.

While there are still multiple avenues worth investigating in improving the performance of the DNN classifier, already the current results demonstrate that it would be possible to replace the current collection of iteration specific boosted decision tree classifiers with a single deep neural network that can classify tracks built in any of the iterations while improving the overall performance.

Part V

Search for the Charged Higgs boson in the $\tau^{\pm}\bar{\nu}_{\tau}$ hadronic decay channel

Chapter 18

Motivation

The charged Higgs boson would be a clear signal of physics beyond the Standard Model of particle physics, as none of the included particles is a charged scalar. A more complex Higgs sector is also part of several theoretical models extending the Standard Model offering a wide spectrum of possible new Higgs bosons like the charged Higgs boson and the doubly charged Higgs boson. An observation of any additional Higgs bosons would be evidence for new physics.

The simplest extensions of the Higgs sector are so called two Higgs Doublet Models (2HDMs) where the additional Higgs field results in a total of five types of Higgs bosons: Two CP-even neutral Higgs bosons h and H ($m_{\mathrm{h}} \leq m_{\mathrm{H}}$), two charged Higgs bosons $\mathrm{H}^{\pm}$ and a CP-odd neutral Higgs boson A. The 2HDM models are classified to different categories based on how the two Higgs doublets are coupled to fermions. The search presented here studies the models where one doublet couples to down type quarks and charged leptons and the other to up type quarks. These are called type II 2HDM models, and theories like the minimal supersymmetric standard model (MSSM) are included in this category.

The dominant process for producing charged Higgs bosons in type II 2HDM models is dependent on the charged Higgs boson mass with respect to the mass of the top quark. The lowest order diagrams for charged Higgs production for the two mass regions and the intermediate region are shown in Figure 18.1. For a light $\mathrm{H}^{\pm}$ scenario where the mass of the particle is less than the difference between the masses of top and bottom quarks ($\mathrm{m}_{\mathrm{H}^{\pm}} \leq m_{\mathrm{t}} - m_{\mathrm{b}}$), the boson can be produced directly through a top quark decay. The heavy charged Higgs boson ($\mathrm{m}_{\mathrm{H}^{\pm}} > m_{\mathrm{t}} - m_{\mathrm{b}}$) is produced in association with the top and bottom quarks. In case the charged Higgs boson mass is near the top quark mass ($\mathrm{m}_{\mathrm{H}^{\pm}} \sim \mathrm{m}_{\mathrm{t}}$) in the so called intermediate region, a non-resonant top quark production mode has a significant contribution to the production and has to be included in the calculation as well.

The charged Higgs boson decays dominantly to a τ lepton and a neutrino in the light mass region. The coupling to leptons is determined by the ratio of the vacuum expectation values of the two Higgs doublets denoted as $\tan\beta$. In the heavy mass region, for high values of the $\tan\beta$ the branching fraction of $\mathrm{H}^{+} \to \tau^{+}\bar{\nu}_{\tau}$ remains significant but the decay channel for $\mathrm{H}^{+} \to \mathrm{t}\bar{\mathrm{b}}$ is dominant across the mass range, with the exception of masses near the top

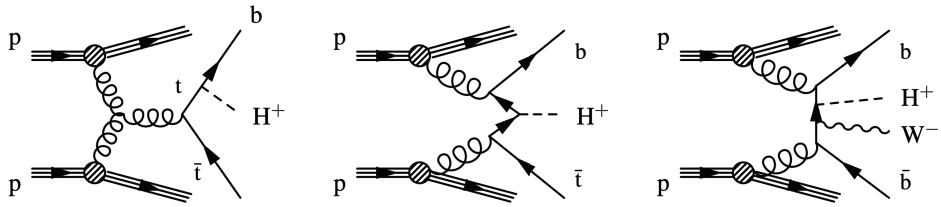


Figure 18.1: The dominant production mechanisms different charged Higgs boson masses. **Left**: The light charged Higgs boson ($m_{H^\pm} \leq m_t - m_b$) can result from a decaying top quark in the double-resonant top quark production. **Middle**: In the heavy charged higgs boson scenario ($m_{H^\pm} > m_t - m_b$) the boson is created in association with a top and a bottom quark in the single-resonant top production. **Right**: In the intermediate region ($m_{H^\pm} \sim m_t$) the non-resonant top quark production process.

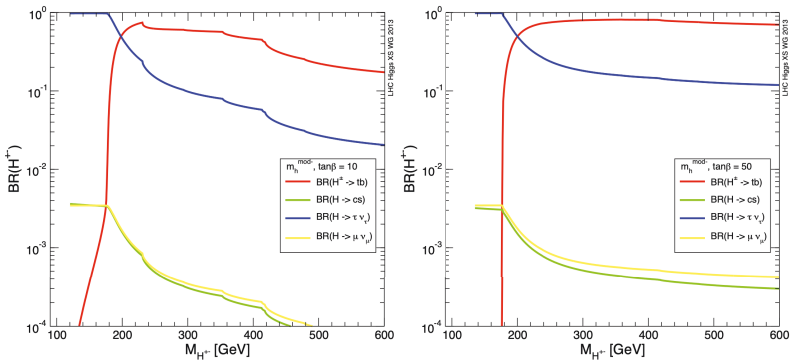


Figure 18.2: Branching fractions for the $H^\pm$ as a function of the $m_{H^\pm}$ in the $m_h^{\text{mod-}}$ benchmark scenario. **Left**: $\tan\beta = 10$. **Right**: $\tan\beta = 50$. Figure from [201]

quark mass. The branching fractions to different end states are presented in Figure 18.2 as calculated for the $m_h^{\text{mod-}}$ benchmark scenario using $\tan\beta = 10$ and $\tan\beta = 50$.

The τ lepton can further decay into a leptonic, semi-leptonic or a hadronic final state. The search presented here focuses on the fully hadronic final state where the tau lepton decays into charged and neutral pions, and a neutrino. This produces a fingerprint of a large amount of missing energy escaping the detector with the neutrino, and multiple jets one of which results from the τ lepton and others from the quarks produced in association with the charged Higgs boson. By identifying the τ-like jet, a transverse mass distribution can be produced for direct measurement of the Jacobian peak caused by the charged Higgs boson in a region enriched by the signal events.

So the overall analysis strategy is as follows: Using the detector fingerprint of the objects created in the charged Higgs boson producing processes described above, a sector of the phase space that is expected to contain significant amounts of signal events compared to

background events is chosen based on simulations. A template fit over the m_T distribution of the events is performed using the side-band regions around each signal mass candidate being studied, representing the expectation for the distribution of background events in the signal region. Based on the number of actual events in the data that is observed in each signal region compared to the expected number of events from the background fit, an upper limit for the charged Higgs boson production is determined.

Chapter 19

Dataset and event selection

The event selection of an analysis is designed to isolate a portion of the phase space where an excess of the collision events of interest can be seen with respect to background events. In case of the charged Higgs boson search from the fully hadronic tau decay channel, the selection takes advantage of the large amount of missing transverse energy resulting from the two neutrinos escaping the detector, the energetic jets from the quarks and the tau, and the relative angles of the jets and the missing energy since they are correlated in case they are resulting from the charged Higgs boson production.

19.1 Data and simulated events

The presented analysis uses the proton-proton collisions collected at the CMS experiment during operations in 2016. The collisions delivered by the LHC took place at a center-of-mass energy of 13 TeV and the dataset collected by the CMS experiment at the high-quality required for physics analyses corresponds to a luminosity of 35.92 fb^{-1}. The average pile-up for the 2016 data is roughly 23 interactions per crossing [202].

The data was included in the Run 2 CMS analysis for the same decay channel in [203] and following the practises of the collaboration the additional analysis presented here will not be unblinding the data in the sensitive signal region but instead the expected sensitivity of the analysis is studied through the use of simulated data, and a comparison with the expected limits using the methods outlined for the published analysis is made.

19.1.1 Signal simulation

The signal samples for the $H^{\pm}$ included in the analysis range from 80 GeV to 3 TeV. The mass points form three distinctive regions: Light $H^{\pm}$ masses (80-160 GeV), heavy $H^{\pm}$ masses (180 GeV-3 TeV) and intermediate $H^{\pm}$ masses (145-200 GeV).

In the light mass range the samples are generated at next-to-leading order (NLO) accuracy using MadGraph5_aMC@NLO v2.3.3 [204] generator.The assumption is that the $H^{\pm}$ is produced through top quark decay ($pp \rightarrow H^{\pm}W^{\mp}b\bar{b}$). The heavy mass range uses the

same accuracy and generator but the production mechanism is assumed to be the associated production with a top quark ($pp \to H^{\pm}tb$) using the four-flavour scheme.

The intermediate mass range samples are generated using leading order accuracy using MadGraph5_aMC@NLO and the model described at [205]. The discrepancy between using LO and NLO samples is corrected for with LO-to-NLO correction factors. The results using the corrected LO samples are found to agree with the results found using the NLO samples in the overlapping regions (145-160 GeV and 180-200 GeV) and as such these regions are analysed using the NLO samples and only the missing intermediate mass points 165, 170 and 175 GeV are done using the LO samples. The intermediate mass samples used are generated using the *no-neutral* samples where the neutral Higgs boson contributions to the charged Higgs boson production ($H \to H^{\pm}W^{\mp}$) are not taken into account since the production mode would introduce model dependency into the results as different 2HDM models give different properties to the neutral Higgs bosons.

All signal sample decays up to $m_{H^{\pm}} = 500$ GeV are modeled using MadSpin [206] and for the rest Pythia 8.212 is used [36].

19.1.2 Background simulation

The background processes consist of $t\bar{t}$, single top, Z/γ^*, W+jets and diboson events. The processes and software used for their simulation are collected in Table 19.1.

Process	Software	Other
$t\bar{t}$	Powheg v2.0	[204, 207–210], FxFx jet matching and merging [211]
Single top (t-channel and tW-production)	Powheg v2.0	[212, 213]
Single top (s-channel)	MadGraph5_aMC@NLO v2.2.2	[212]
Z/γ^*	MadGraph5_aMC@NLO v2.2.2	LO, up to 4 noncollinear parton final state [214]
W+jets	MadGraph5_aMC@NLO v2.2.2	LO, up to 4 noncollinear parton final state [214]
Diboson (WW, WZ, ZZ)	Pythia 8.212	[36]

Table 19.1: Software used for background processes. For all $t\bar{t}$ and single top samples the $m_t = 172.5$ GeV.

After the hard interaction is simulated all samples including the signal sample are processed through Pythia 8.212 for parton showering, fragmentation and tau lepton decay simulation. The underlying event tune is set to cuetp8m2t4 [215] and the NNPDF3.0 parton distribution function [216] is used for all simulated samples. In order to have the simulated events presented in a uniform fashion to the reconstruction algorithms used to process the collected real data, a detector simulation is run on all simulated samples using Geant4 v9.4 simulator [166, 217]. The additional events corresponding the pile-up collisions present in the detector

during data taking are modeled using minimum bias collision events generated with PYTHIA and mixing them in with the simulated hard interactions.

19.2 Statistical analysis

In order to interpret the data that got through the selection, a null hypothesis must be first determined that the results can then be compared against. When searching for a new particle beyond the Standard Model the null hypothesis becomes *the new particle does not exist* and the measured data should match with what would be produced by the Standard Model processes. The summary statistic used to do the hypothesis testing is the transverse mass distribution of the events passing the selection.

The amount of signal is represented by a *signal strength modifier* μ. Given s (b) as the expected event yield for signal (background) events, the expected total yield is $\mu s + b$. For the charged Higgs boson analysis where the production cross section σ and the branching fraction to the final state $\mathcal{B}$ are not predicted, the expected signal yield is normalized using $\sigma = 1$ pb, $\mathcal{B} = 100\%$. The goal of the analysis is to set an upper limit on the value of μ based on the collected data which in turn can be interpreted as the limit on $\sigma\mathcal{B}$. The method used by the CMS and ATLAS collaborations [218] is used here.

Having n_i observed events in a bin i when the expected event yield for the bin is given as $\mu s_i + b_i$ follows the Poisson probability distribution. The combined probability over all bins i is given by

$$\mathcal{L}(\{n_i\}|\mu) = \prod_i \frac{(\mu s_i + b_i)^{n_i}}{n_i!} e^{-\mu s_i + b_i} \tag{19.1}$$

where $\{n_i\}$ is the collection of bins with n_i being the number of events in bin i. The value of μ that maximizes this likelihood function for the observed data is denoted as μ_{ML}. A test statistic $\tilde{q}_\mu$ is defined as the likelihood ratio

$$\tilde{q}_\mu = -2 \ln \frac{\mathcal{L}(\{n_i\}|\mu)}{\mathcal{L}(\{n_i\}|\mu_{\mathrm{ML}})}, \tag{19.2}$$

based on the *modified frequentist* criterion [218] also known as the CL method. The value of $\tilde{q}_\mu$ is minimized with two constraints. The value $\mu \geq 0$ ($\mu_{\mathrm{ML}} \geq 0$) meaning that any signal has to show up in the measurement as additional events instead of subtracting the events. Additionally $\mu_{\mathrm{ML}} \leq \mu$ meaning that the test is to see whether the signal is at least as strong as expected by the hypothesis.

The expected distributions for $\tilde{q}_\mu$ are produced by generating pseudoexperiments from sampling the Poisson distribution with a mean of $\mu s_i + b_i$. The pseudoexperiments are performed using both the null hypothesis ($\mu = 0$) and the signal hypothesis ($\mu = \mu_{\text{sig}}$). After normalizing the distributions from the pseudoexperiments to unity, they are probability density functions for $f(\tilde{q}_\mu|\mu = \mu_{\text{sig}})$ and $f(\tilde{q}_\mu|\mu = 0)$ that can be used to produce p-values for the observation.

P-value represents the probability for obtaining at least as extreme results as the observation is from a statistical hypothesis test under the assumption that the null hypothesis is true. When the p-value is smaller than some threshold α, the null hypothesis is said to be excluded at a confidence level of $1-\alpha$. The p-value for a signal of strenght μ_{sig} (or larger) being present in the data and giving a value of at least $\tilde{q}_\mu$ is given by

$$p_\mu = \int_{\tilde{q}_\mu^{\text{obs}}}^{\infty} f(\tilde{q}_\mu|\mu = \mu_{\text{hyp}}) d\tilde{q}_\mu. \tag{19.3}$$

Conversely the p-value for $\tilde{q}_\mu$ being less than the observed value when there is no signal present is calculated as

$$p_b = \int_{-\infty}^{\tilde{q}_\mu^{\text{obs}}} f(\tilde{q}_\mu|\mu = 0) d\tilde{q}_\mu. \tag{19.4}$$

The CL method uses both p_μ and p_b to determine whether the null hypothesis is excluded or not, by defining

$$\text{CL}_\text{S} = \frac{p_\mu}{1 - p_b} \tag{19.5}$$

as the variable to use instead of the p-value. If the signal hypothesis μ_{sig} gives CL_S value of α, it is said that signals stronger than μ_{sig} are excluded at (1-α) confidence limit. The limit calculation then proceeds by finding the value of μ that yields $\text{CL}_\text{S} = 0.05$ denoted as $\mu_{95\%}$ and the value $\mu_{95\%}\sigma\mathcal{B}$ is said to be the exclusion limit for the product of cross section and branching fraction of the particle.

As a counterpart to the observed exclusion limit is the expected exclusion limit. They are produced by performing a series of pseudoexperiments as described above using Poisson distributions with means n_i from the yields. These results are then ordered by $\mu_{1-\alpha}$ and normalized to produce a probability distribution function. From this function the median

value as well as standard deviations from it can be calculated. In order to be able to estimate the sensitivity of an analysis before the data is unblinded, the expected limits can be produced using an *Asimov dataset* to give the bin yields n_i instead of the real data. In this dataset the values n_i are set as the expectation $\mu_{\text{sig}} s_i + b_i$

19.3 Selection flow

The selection flow is shown in Table 19.2 for both the online and offline selections. This section presents selections 1-9 in detail, while the last selection using a deep neural network is discussed in Chapter 20.

	Selection	Description
1	$\tau_h + \vec{p}_{\text{T}}^{\text{miss}}$ trigger	Roughly signal-like events chosen with trigger
2	Data quality filters	Events with spurious $\vec{p}_{\text{T}}^{\text{miss}}$ are rejected
3	τ_{h} identification	At least one τ_{h} ($p_{\text{T}} > 50$ GeV, $\lvert\eta\rvert < 2.1$)
4	Lepton veto	Removes events with isolated electrons or muons
5	Jet selection	At least three jets ($p_{\text{T}} > 30$ GeV, $\lvert\eta\rvert < 4.7$)
6	b-jet selection	At least one b-tagged jet ($p_{\text{T}} \geq 30$ GeV, $\lvert\eta\rvert < 2.4$)
7	$\vec{p}_{\text{T}}^{\text{miss}}$	Type-I corrected $\vec{p}_{\text{T}}^{\text{miss}} > 90$ GeV
8	Angular selection	Reduce jet$\rightarrow \tau_{\text{h}}$ background with jet and $\vec{p}_{\text{T}}^{\text{miss}}$ directions
9	R_τ categorization	Measure events with $R_\tau > 0.75$ and $R_\tau \leq 0.75$ separately
10	DNN classifier	Classifier to separate signal and background events

Table 19.2: The selection flow designed to choose a region of phase space enriched with charged Higgs boson like events. Selections 1 is performed online during data collection. Selection 2 removes events with problems occuring during data taking resulting in excessive $\vec{p}_{\text{T}}^{\text{miss}}$. Selections 3-5 are the *baseline* selection used in determining the transfer factors for data-driven background estimation. Selections 6-8 aim to remove background from events where a jet is misidentified as a τ_{h}, selection 9 reduces the amount of background events with a genuine τ_{h} and selection 10 aims to further enhance the signal-background ratio by reducing all backgrounds remaining.

19.4 Trigger selections

As the topology of the signal-like events contains an energetic τ_{h} and large missing transverse energy, both of these are required at the trigger level already. At the HLT level the τ_{h} is reconstructed using a fast cone based algorithm that combines information from calorimeter deposits and pixel tracks to verify a jet candidate that is suitably isolated from charged tracks to match the signature of a τ_{h} [219]. The τ_{h} candidates at the HLT are required to have $p_{\text{T}} > 50$ GeV with a leading charged track with $p_{\text{T}} > 30$ GeV. Due to the pixel tracker coverage τ_{h} candidates can be reconstructed only for $|\eta| < 2.1$.

The $\vec{p}_{\text{T}}^{\text{miss}}$ at HLT is determined as the negative vector sum of the transverse energy measured by the calorimeter. The missing transverse energy is required to be $\vec{p}_{\text{T}}^{\text{miss}} > 90$ GeV.

Combined together the $\tau_h + \vec{p}_T^{miss}$ trigger selects events compatible with the hypothesis of containing a charged Higgs boson decaying into the fully hadronic tau channel.

19.5 Data quality filters

Detector malfunctions and failures in reconstructing the objects in the collision can result in mismeasured $\vec{p}_T^{miss}$. A set of data quality filters are applied to remove events likely to contain erroneous energies due to calorimeter malfunctions, wrongly reconstructed high p_T objects or interference from collisions upstream of the detector as described in [144].

19.6 Baseline selections

The baseline selections set the requirements for the necessary objects for the signal fingerprint to be present in the event. These selections are loose by design to avoid significant suppression of the fake τ_h background resulting from jets $\rightarrow \tau_h$. This dominant background is measured after the baseline selections from data using the method described in Section 21.1 while the genuine τ_h contribution will be estimated from simulations.

19.6.1 τ_h identification

In the offline selection same thresholds of $p_T > 50$ GeV and $p_T^{ldg.trk} > 30$ GeV are applied to the reconstructed τ_h candidate. Additionally the candidates are required to pass the loose working point of the MVA discriminant, having an identification efficiency of $\approx 50\%$ with a misidentification rate of 3×10^{-3}. Only one-pronged τ_h candidates where the τ_h decay products contain only one charged pion are accepted into the analysis. Rejecting the three-pronged candidates would require a completely separate data-driven background estimation and only produces a small improvement in sensitivity. In case there are more than one τ_h candidates passing the selection, the one with the largest p_T is chosen for the analysis.

19.6.2 Lepton veto

Isolated electrons (muons) with $p_T > 15(10)$ GeV and $|\eta| < 2.5$ are searched for and events containing them are rejected. Both leptons are identified using the loose working point of the corresponding discriminators. The isolation condition is computed by adding the p_T of all PF candidates inside an isolation cone around the lepton and requiring it to be less than 40% of the lepton p_T. This removes events where a $W^{\pm}$ resulting from a top quark decays into a lepton and smears the m_T distribution due to the additional neutrino. Additionally the selection makes the events in the final analysis orthogonal to the analyses studying the leptonic charged Higgs boson final states allowing for statistical combination of the results of the different searches.

19.6.3 Jet selection

To account for the topology of the final state in the search channel at least three jets are required to be present in the event each with $p_{\mathrm{T}} > 30$ GeV, $|\eta| < 4.7$ and separation of $\Delta R > 0.5$ from the hadronic tau. Additionally jets are required to satisfy a loose set of Jet ID criteria used in the CMS described at [220]. The jets used in the analysis are reconstructed from particle flow candidates with the anti-k_{T} [167] algorithm using distance parameter $R = 0.4$.

19.6.4 B-jet selection

One of the jets in the event is required to be identified to originate from a b-quark, since it is an expected by-product of the charged Higgs boson production. The b-jets are identified using a *combined secondary vertex* (CSV) algorithm [221], which takes advantage of the displacement of the jet origin due to the finite lifetime heavy-flavour hadrons associated with the hadronization of the b-quark.

These hadrons can travel between few millimeters and up to a centimeter within the detector before decaying and the charged tracks from the decay can be traced back to the displaced vertex. The CSVv2 algorithm is based on a multivariate discriminant predicting how b-jet like the object is based on the input variables and the working point in the analysis is chosen so that the misidentification probability of assigning jets originating from light-flavour quarks or gluons as b-jets is around 1% while the efficiency for tagging genuine b-jets is 65% based on simulation. This selection limits the b-jet candidates to $|\eta| \leq 2.5$ as the tracker is used in tagging the b-jets.

19.6.5 Missing transverse momentum selection

Since the two neutrinos present in the final product of the decay are expected to carry a significant amount of energy, a large amount of $\vec{p}_{\mathrm{T}}^{\,\mathrm{miss}}$ is required in the event selection. This is computed using as the negative vector sum of all particle flow candidates in the event and corrected by propagating the jet energy corrections giving the *Type-I* corrected $\vec{p}_{\mathrm{T}}^{\,\mathrm{miss}}$ as

$$\vec{p}_{\mathrm{T}}^{\,\mathrm{miss}} = \vec{p}_{\mathrm{T}}^{\,\mathrm{miss,\ uncorr.}} - \sum_{\mathrm{jets}} (\vec{p}_{\mathrm{T}}^{\,\mathrm{corr.}} - \vec{p}_{\mathrm{T}}^{\,\mathrm{uncorr.}}), \tag{19.6}$$

where $\vec{p}_{\mathrm{T}}^{\,\mathrm{miss,\ uncorr.}}$ is the negative vector sum of the PF candidates, $\vec{p}_{\mathrm{T}}^{\,\mathrm{corr.}}$ is the corrected jet transverse momentum and $\vec{p}_{\mathrm{T}}^{\,\mathrm{uncorr.}}$ is the jet transverse momentum before corrections. Energy corrections to other reconstructed objects are considered negligible in comparison to jet energy corrections. The same threshold of $\vec{p}_{\mathrm{T}}^{\,\mathrm{miss}} > 90$ GeV is set for the offline selection as required by the HLT trigger.

19.6.6 Angular selection

The background events resulting from jets incorrectly tagged as hadronic tau candidates, jet $\rightarrow \tau_{\mathrm{h}}$, is largely from QCD multijet events where the large $\vec{p}_{\mathrm{T}}^{\,\mathrm{miss}}$ value is due to misreconstructed jet momenta causing an imbalance between the energies of a jet pair. In this type of events, the mistagged τ_{h} candidate and the $\vec{p}_{\mathrm{T}}^{\,\mathrm{miss}}$ are back-to-back due to both resulting from the same dijet pair and this leads to a large m_{T} value being assigned to the event.

This background can be reduced by taking advantage of the fact that the angle between the τ_{h} and $\vec{p}_{\mathrm{T}}^{\,\mathrm{miss}}$ in this scenario is large and combining this information with the fact that the $\vec{p}_{\mathrm{T}}^{\,\mathrm{miss}}$ results from a mismeasurement of one of the jets in the event so by construction it should be collinear with one of the jets in the event as long as it gets identified. As this is not the case for the events where the $\vec{p}_{\mathrm{T}}^{\,\mathrm{miss}}$ resulting from the neutrino has no reason to be collinear with the jet direction.

The selection is done by cutting on a variable $R_{\mathrm{bb}}^{\mathrm{min}}$ which gets a high value when τ_{h} and $\vec{p}_{\mathrm{T}}^{\,\mathrm{miss}}$ are back-to-back and one of the leading jets is collinear with $\vec{p}_{\mathrm{T}}^{\,\mathrm{miss}}$:

$$R_{\mathrm{bb}}^{\mathrm{min}} = \min_{n} \left(\sqrt{(180^{\circ} - \Delta\phi(\tau_{\mathrm{h}}, \vec{p}_{\mathrm{T}}^{\,\mathrm{miss}}))^2 + \Delta\phi(\mathrm{jet}_n, \vec{p}_{\mathrm{T}}^{\,\mathrm{miss}})^2} \right), \tag{19.7}$$

where the index n runs over the three selected jets in the event, $\Delta\phi(\tau_{\mathrm{h}}, \vec{p}_{\mathrm{T}}^{\,\mathrm{miss}})$ is the angle between τ_{h} and $\vec{p}_{\mathrm{T}}^{\,\mathrm{miss}}$ and $\Delta\phi(\mathrm{jet}_n, \vec{p}_{\mathrm{T}}^{\,\mathrm{miss}})$ is the angle between the nth jet and $\vec{p}_{\mathrm{T}}^{\,\mathrm{miss}}$. When a jet is collinear with $\vec{p}_{\mathrm{T}}^{\,\mathrm{miss}}$ the second term in the square root goes to zero so by choosing the minimum among the jets the selection considers the jet that is most collinear with $\vec{p}_{\mathrm{T}}^{\,\mathrm{miss}}$. Based on optimization $R_{\mathrm{bb}}^{\mathrm{min}} > 40^{\circ}$ is chosen as cut value that reduces significantly QCD multijet background without throwing out too much of the signal events. A visualisation of this selection in the $\Delta\phi(\tau_{\mathrm{h}}, \vec{p}_{\mathrm{T}}^{\,\mathrm{miss}}), \Delta\phi(\mathrm{jet}_n, \vec{p}_{\mathrm{T}}^{\,\mathrm{miss}})$-plane is presented in Figure 19.1.

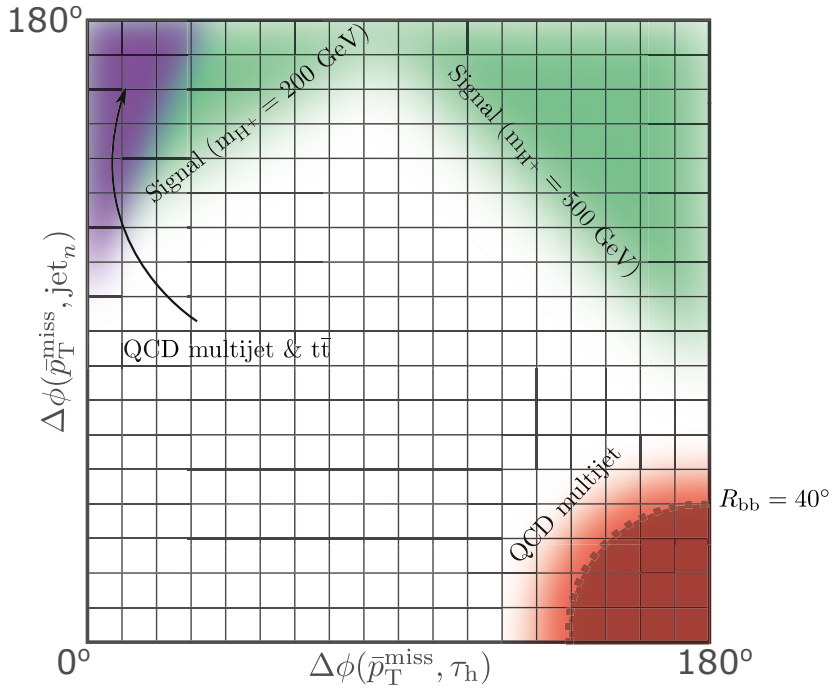


Figure 19.1: Visualization for the intended effect of the R_{bb}^{min} cut. The region contained in $R_{bb}^{min} < 40°$ is mostly populated by the QCD multijet events where a jet mismeasurement is the cause of the large $\vec{p}_T^{miss}$ value.

19.6.7 Deep neural network classifier

A deep neural network is used in order to leverage the correlations among the variables for maximising the separation between the signal and background. The inputs used in the network are the well established variables also utilized in the selection cuts described above, but the network trained to combine the information from the variables in order to perform a more refined cut on the dataset based on the output discriminator value.

An additional challenge for the network training comes from the way the limit calculation is performed on the selected events. This is done based on a chosen variable of interest, in our case the transverse mass m_{T}, and it relies on an interpolation of the background distribution for the values of the variable of interest in the signal region based on sideband regions around it. For this method to produce good results the selections performed on the dataset should not distort the background distribution in a non-uniform way. As was discussed in more detail in Section 10 the classifier should be decorrelated from the variable of interest for this reason and this is done using the distance correlation method. Training and performance of the deep neural network classifier is presented in Chapter 20.

19.6.8 Event categorization

After all selections the subset of events is separated into two categories based on the helicity correlations of the τ lepton which cause the taus from $\mathrm{H}^{\pm}$ to dispense a larger fraction of their momenta into the leading charged track on average than the background events. The categorization is done based on $R_\tau = p_{\mathrm{track}}/p_{\tau_{\mathrm{h}}}$, where p_{track} is the three-momentum of the leading charged track and $p_{\tau_{\mathrm{h}}}$ is the three-momentum of the tau candidate.

The reason for using this categorization instead of making a cut on the R_τ value is due to the interplay between different uncertainties and the signal purity. While requiring a high R_τ value from the events improves signal-to-noise ratio, it can increase the statistical uncertainty in the high m_{T} regions negating the benefits. In order to get the most out of this selection, the dataset is split into two orthogonal sets based on a selection $R_\tau > 0.75$ and the limits calculated from the two sets are combined. This leads to improved sensitivity of the analysis across the range instead of just improving the sensitivity in one region of the m_{T} spectrum and deteriorating it in another.

Chapter 20

Deep neural network classifier for events

Deep neural networks have been demonstrated to be highly effective in discriminating between signal and background from high dimensional inputs. The other event selections in the analysis described using often single variables and making a decision to cut out some of the events based on some threshold value of that variables. This type of an approach has the weakness of missing out on taking advantage of correlations between these input variables to improve the selection efficiency. In some scenario for example there could be a signal enriched region of phase space where the τ_{h} has a relatively low p_{T} value and the b-jet p_{T} has a large value. The analysis workflow described above would end up discarding all of this signal if τ_{h} p_{T} is below the cut-off value. A more sophisticated algorithm that is able to look at both of these values simultaneously and take advantage of this hypothetical correlation between them could instead choose to include this region into the final set of events. One such family of algorithms are the deep neural network classifiers that can learn these correlations between variables from looking at the simulated events.

In the work presented here the deep neural network is studied as a method to improve upon the existing analysis workflow that can further improve the sensitivity of the selection by relaxing some of the other selection conditions in order to better take advantage of the correlations among the input variables to perform a multidimensional analysis and separate the signal events from the background events based on training on simulated collision events. The final results will be presented as the confidence limits derived from the m_{T} distribution and as such extra steps are taken to ensure minimal distortion of the distribution from this new selection by using decorrelation methods during the training of the classifier. This novel analysis will be performed on the CMS 2016 dataset and it will be compared to the baseline selection performed on the same dataset using otherwise identical event selection with the exception of the DNN classification step.

20.1 Training dataset

For the purpose of training the classifier the simulated dataset is split in half based on the event identification number and two separate classifiers are trained with everything else kept identical: One is trained on odd numbered events and deployed to be used on the even numbered events and vice versa for the other. This enables the full dataset to still be used in the actual analysis without biasing the results with a classifier that would have seen some of the events during the training process. This is important since having to discard portion of the analysis dataset due to it having been used in the training would increase the statistical uncertainty in the final analysis result.

The training set is chosen by using the baseline selections and the b-jet selection in order to ensure that all the input variables are well defined for all the events as two of them are with respect to the b-jet. The other selections like the angular cut and $\vec{p}_{\mathrm{T}}^{\,\mathrm{miss}}$ requirements are not enforced in order to increase the training statistics. Additionally since the analysis is split into two categories based on the R_τ value it is important to show the classifier examples from both categories.

To ensure balanced representation of different signal masses as well as signal and background categories the set is subsampled without replacement so that an equal number of background and signal events are present. The final statistics for the training dataset contain $O(10^6)$ events in 1:1 ratio between signal and background, out of which 10^4 are used in the validation dataset during training to monitor for over-fitting and control the learning rate schedule.

20.2 Input variables and preprocessing

The input variables are chosen to include relatively high-level variables known to contain useful information for discriminating between the signal and background events. These are mostly the same variables already used in the other selection steps during the analysis workflow, but the neural network is trained to look at all of them at the same time and make more informed classification based on learning correlations between the input values.

The nine variables used as inputs are described in Table 20.1. With the similar considerations as were presented in the discussion about the track classifier network in Section 17.2, a preprocessing scheme is also applied in the charged Higgs boson event classifier. In order to make the network training more stable and consistent all input variables are bound within the same numerical range between minus one and one. For each feature this is achieved by subtracting the smallest element and dividing by the largest element present in the training set, multiplying the result by two and subtracting one in a process known as min-max scaling:

$$\bar{x}' = \text{min-max}(x) = 2 \cdot \frac{\bar{x} - \min(\bar{x})}{\max(\bar{x}) - \min(\bar{x})} - 1, \quad \bar{x}' \in [-1.0, 1.0]. \tag{20.1}$$

Variable	Description
$p_{\mathrm{T},\tau_{\mathrm{h}}}$	Transverse momentum of the τ_{h} candidate
$\vec{p}_{\mathrm{T}}^{\,\mathrm{miss}}$	Missing energy in the event
$R_{\tau_{\mathrm{h}}}$	Leading charged track energy fraction for τ_{h}
$p_{\mathrm{T,\ b\text{-}jet}}$	Transverse momentum of the b-jet
m_{T}	Transverse mass between τ_{h} and $\vec{p}_{\mathrm{T}}^{\,\mathrm{miss}}$
$\Delta\Phi(\tau_{\mathrm{h}}, \vec{p}_{\mathrm{T}}^{\,\mathrm{miss}})$	Angle between τ_{h} momentum and $\vec{p}_{\mathrm{T}}^{\,\mathrm{miss}}$ direction
$\Delta\Phi(\tau_{\mathrm{h}}, \text{b-jet})$	Angle between τ_{h} momentum and b-jet momentum
$\Delta\Phi(\text{b-jet}, \vec{p}_{\mathrm{T}}^{\,\mathrm{miss}})$	Angle between b-jet momentum and $\vec{p}_{\mathrm{T}}^{\,\mathrm{miss}}$ direction
m_{true}	True mass (randomly chosen mass) for signal (background)

Table 20.1: The DNN input variables. During preprocessing for the first five variables (green) the natural logarithm of $1 + x$ is taken where x is the variable. For the three angles (red) just the absolute value is of the variable is used. True mass (blue) is the simulated mass point value for signal samples and a randomly chosen simulated signal mass point for the background during training, as detailed below for parametrized neural network.

Additionally for features like p_{T}, $\vec{p}_{\mathrm{T}}^{\,\mathrm{miss}}$ and m_{T} where the samples are spread out over many orders of magnitude, an additional logarithmic scaling is applied to the entries to make the distribution less sparse.

20.3 Parametrized neural networks

A notable detail is that one of the input variables is the transverse mass computed from the measured momentum of the τ_{h} candidate and the missing energy in the event. This transverse mass is not known for the charged Higgs boson and for that reason the analysis is performed over a range from 80 GeV up to 3 TeV where signal samples are simulated at different candidate masses. This scenario requires some precautions to be taken in order to train a classifier that is able to achieve good performance across the mass range being studied. Since the transverse mass of the event will have correlations with the other input variables as well, a naively trained classifier might perform well on some mass point of the signal samples but fail catastrophically on others.

This specific aspect of the learning task was studied by P. Baldi *et al.* [222] and they presented the concept of parametrized neural networks that would allow a single network to learn to smoothly interpolate between different hypothetical masses of the signal using only a finite set of simulated signal masses. Before parametrized networks, there were a few other approaches. One could train multiple neural networks each of which was responsible for some mass range which was impractical as it required more effort to train and store multiple networks. This could also lead to undesirable discontinuities at the edges of the mass ranges where two different networks are providing the predictions. Alternatively one could just use a mixture of samples generated with different masses to train a single classifier to be used for the whole mass range, but this could lead to reduced sensitivity of the classifier across the range compared to using multiple classifiers. The third option was to use just a single mass

point for training and hope the classifier extrapolates to the other points without issues.

In parametrized neural networks the idea is to add one or more parameters $\bar{\theta}$ to the input describing some larger context about the entry, in this case the true mass value used for the simulation. With this additional parameter in place they demonstrated that the parametrized neural network trained with realistic signal samples generated from a finite set of mass points is able to match the performance on an unseen mass point with a neural network classifier that was trained specifically on samples generated with that signal mass. In other words the parametrized network provides as good a performance across the mass range it was trained for as can be expected.

This method is also employed with the classifier presented here. The true mass is included as an additional input to the training samples for the signal. For background events during training and for all events when deploying the model, this true mass parameter is replaced by a randomly drawn value from the distribution of true masses used in the training as was done in the original paper. It is noted that the robustness of this method relies on the neural networks ability to generalize these true mass values and in having sufficient training statistics.

20.4 Decorrelation from transverse mass

As discussed in Chapter 10 there might be certain variables the classifier's decisions should be decorrelated from. In the analysis presented here the final result is computed by fitting the background event shape from the side-band regions around the signal region using the m_{T} distribution of the events, so the event selection should avoid distorting the background shape in unexpected ways to ensure a good fit around the mass points used for signal simulation. This topic has been researched in the context of high energy physics in multiple sources during the recent years as deep neural network classifiers have become more popular in physics analyses and object identification [77, 82, 223–226] indicating the importance of this topic going forwards.

The decorrelation methods considered include using hand-crafted input features where the correlations between the inputs and variable of interest have been removed, planing the input distributions to be flat with respect to the variable of interest, training an adversarial neural network to drive the classifier towards a solution that is uncorrelated with the variable of interest and augmenting the loss function with an additional term that will favour solutions where output has no correlation with the variable of interest. During the research presented here all of the aforementioned methods have been tested in the context of producing a deep neural network classifier for the charged Higgs boson search in this decay channel, with their merits and problems briefly discussed here before diving into the chosen method in more depth.

Planing: Simply reweighting the training samples with respect to transverse mass is the simplest method both conceptually and to be implemented. The idea is that if the classifier sees the ratio of signal samples and background samples as flat along the variable of interest it cannot at least directly use a selection cut in that variable to improve this ratio. However this

reweighting is not guaranteed to prevent the classifier to shape the distribution by learning some higher order correlations from the other variables resulting in a output that is ultimately correlated with the variable of interest.

Adversarial training: The adversarial network training has shown impressive results in generative adversarial networks and the like, and it is able to perform the decorrelation task in our context as well. The upside of the method is that the complication of having two separate networks, one classifier and one adversary, is only necessary during the training phase after which only the classifier needs to be deployed. Some success was originally achieved with the adversarial training in this context as well, but the cost in time spent trying and retrying the training with different parameters and making consistently reproducible results turned out to be a difficult task. The main issue is the inherent instability of the training task where two networks are trained in tandem with somewhat opposing goals, which is well known and recognized in the field. This makes implementing the training phase difficult and ultimately problem dependent so that no ready made recipes are applicable.

Augmenting loss: The third method of augmenting the loss function with a suitable term that drives the minimum towards solutions where the predicted output value has no correlation with transverse mass. The appeal of the method is the minimal effort in implementing it as it only requires changing the loss while the rest of the training and network architecture can remain the same. The difficulty is to find a suitable loss term that can be estimated reliably during training, is differentiable for backpropagating to the network weights and describes the linear and nonlinear correlation between the output and variable of interest. Such loss was presented in [82] and was discussed in detail in 10.2 and it is the approach chosen here due to empirical ease of use while achieving the desired result. Additional benefit is that the degree of decorrelation can be adjusted with just a single hyperparameter used during training, so that the method is easy to tune for the use case.

20.5 Training the classifier

As the classifier will be applied as an additional selection after the other cut based selections it should be trained accordingly focusing on events that will pass the other selections. The training data is preprocessed by passing them through the analysis selection workflow and storing the selected events for training. The original selection cuts are slightly relaxed in order to enhance training statistics and avoid biasing the classifier predictions from edge effects on the input variable values.

The loss function used for the classifier is a binary crossentropy loss with an additional distance correlation term

$$L(\mathbf{x}, y) = \mathrm{BCE}(\mathbf{x}, y) + \lambda \cdot \mathrm{dCorr}^2(\mathbf{x}, y) \tag{20.2}$$

The training events are sampled to contain equal amount of signal samples for every mass

point included in the analysis. Additionally the number of signal and background events are subsampled to be equal. The number of events representing different backgrounds are not balanced as the cut based event selections heavily affect the background composition and adjusting it for the training could bias the network.

As per the recommendations in [82] the training minibatch sizes tested are a unusually high values. This helps the distance correlation term stability as it is calculated for individual minibatches and it needs to get an accurate enough representation of the whole training sample in order to provide good gradients for the updates. The value of the hyperparameter λ is searched experimentally to obtain a reasonable trade-off between the level of decorrelation and classification performance.

In order to maximise the amount of statistics available for training without reducing the statistics available for running the analysis workflow the collected training dataset is split into two based on the event ID number assigned to each event when simulated and two independent classifiers are trained with otherwise identical hyperparameters: One is trained on events with odd event ID and used to evaluate the events with even event ID and vice versa for the second classifier. This approach is known as k-fold cross validation where value $k = 2$. It could be taken further by for example using $k = 10$ resulting in 10 separate networks each of which would trained with 90% of the dataset and used to analyse the remaining 10%. However $k = 2$ is chosen here due to simplicity as the higher number of classifiers slows down both the training the networks and running the analysis. When using k-folding the statistics available for the analysis will not be reduced nor will there be bias introduced by evaluating events the classifier has already seen during training. This is just a method for ameliorating the lack of statistics otherwise available, in an ideal case there would be the possibility to simulate an arbitrary amount of samples for the training that would be separate from the datasets used in the analysis.

The effect of using the augmented distance correlation loss function compared to not including it in the loss term ($\lambda = 0$) is studied in Figure 20.1. In this demonstration signal samples are included only from mass points $m_{\mathrm{H}^\pm} = 180$ GeV and $m_{\mathrm{H}^\pm} = 500$ GeV are included to highlight the effect of learning the mass. Including all the signal mass points will naturally flatten the signal sample distribution with respect to the reconstructed mass, corresponding to planing that was presented as one of the decorrelation methods. In the actual training the samples from all the signal points are included as any additional decorrelating effects from planing are simply considered as positive.

The output distribution for background as a function of reconstructed m_{T} is shown on the top row and the distribution of the classifier output values and the mean reconstructed m_{T} per bin on the bottom row. The columns display different values for the parameter λ controlling the contribution of the distance correlation term to the loss function. The most dramatic effect of the regularization added by the distance term can be seen in the low end of the mass spectrum where there is very little signal in the training set and the $\lambda = 0.0$ classifier learns this feature, leading to significant distortion to the original shape of the m_{T} distribution in that region. Additionally the mean m_{T} as a function of the DNN output value has a clear rising trend seen in the bottom plot, indicating that the classifier ends up considering higher m_{T} values to be more signal like.

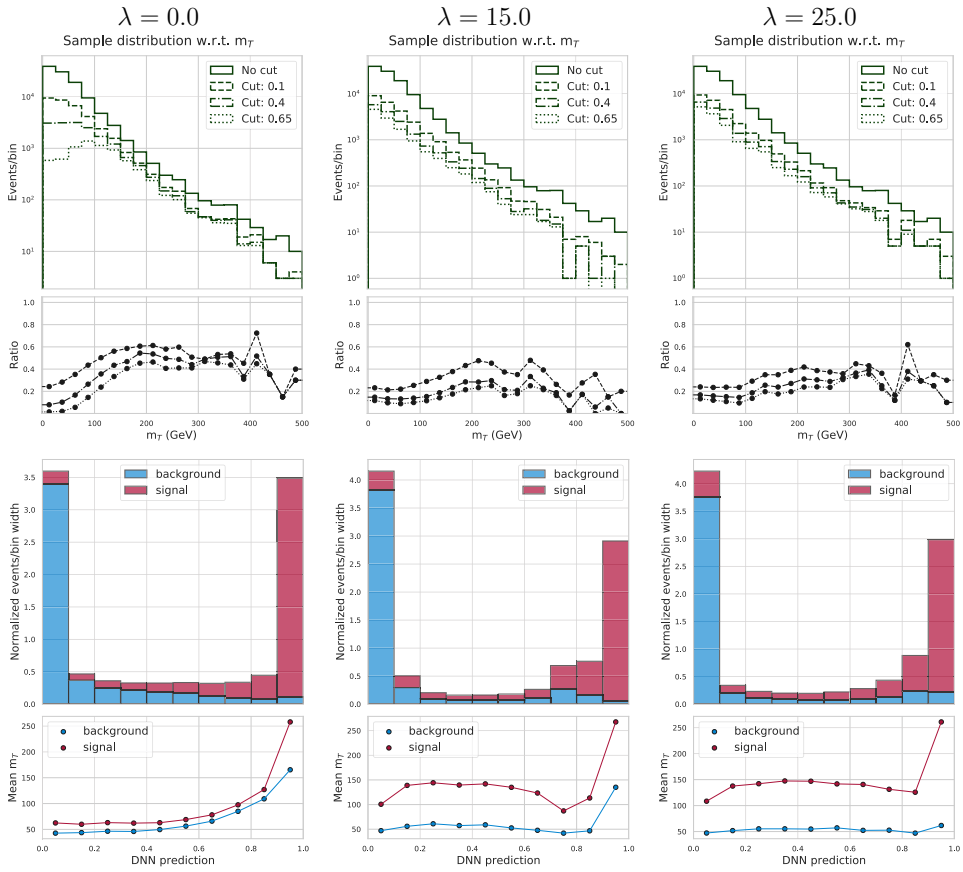


Figure 20.1: The effect of the hyperparameter λ controlling the impact of the distance correlation term in the loss function. Values $\lambda = 0.0$, $\lambda = 15.0$ and $\lambda = 25.0$ are shown in columns from left to right. **Top**: The distribution of the background events in the test set with respect to reconstructed m_{T} and how different cut values based on the DNN output will distort the shape. **Bottom**: The distribution of DNN output values for background and signal events and the mean reconstructed m_{T} per bin.

Increasing the value of λ seems to correct both of these features to an extent. It is worth emphasising that the signal samples do not contribute to the distance correlation loss term so the fact that the highest DNN output bin in the bottom row plots seems to have a higher mean m_{T} for signal samples is expected and desirable. It is also good to note that while the augmented loss has a significant effect in reducing the amount of correlation between the background m_{T} and DNN output there still seems to be some amount of correlation with the transverse mass as the $\lambda = 25.0$ column demonstrates. Some of the jaggedness in the change in the m_{T} shape at high end of the spectrum is explained by the small statistics but still the mean DNN output value for background seems to be higher in $[200, 300]$ GeV region than it

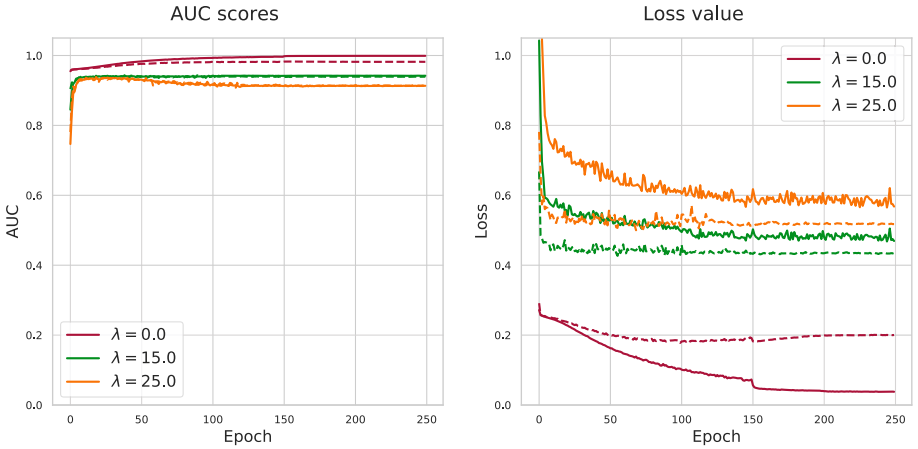


Figure 20.2: Training performance of the different models. Dotted lines represent the quantities computed on the validation dataset that is not directly used in the optimization. The learning rate of the model is decreased after 70 epochs without decreasing the validation loss to force convergence.**Left**: The AUC scores for the same model trained with λ values. Increasing the λ trades the model performance in classification to reducing the correlation with the variable of interest m_{T}. **Right**: Loss trajectories for the models with different λ values.

is for example in the $[0, 100]$ GeV range.

While the original paper noted that the training seems fairly stable with a large range of λ values in the problems they studied, the same does not seem to hold for this problem setting in particular. Depending on other hyperparameters like the batch size, there seems to be a quick transition in the learning dynamics where the network ends up optimizing only the correlation term and ignoring the classification task altogether, predicting only a singular output value for any inputs. This will reduce the correlation term to zero but produce a worthless classifier.

In Figure 20.2 the loss trajectory and the AUC score of the classifier for the different λ values are shown. As expected deterring the use of m_{T} or variables correlated with it in the decision making ends up reducing the AUC score describing the classifier's discriminative capabilities. So the size of parameter λ is a trade-off between classification performance and preserving the background distribution shape.

After these initial tests validating the augmented loss has the desired effect on the classifier performance all mass points are included in the signal samples and the training of the classifier is performed. After initial hyperparameter search value of $\lambda = 15$ and minibatch size of $N = 1024$ is set for the model, and a hyperparameter search through using the hyperband algorithm [197] is performed on a selection of model architectures and hyperparameters. The resulting architecture and parameters are described in 20.6.

For the training set each signal mass point is sampled in equally without replacement and the number of background and signal events are sampled in 1:1 ratio without replacement. Different background event categories are not sampled separately, instead the background events are present in the same ratio as they have passed the baseline selection cuts.

20.6 Neural network architecture

Since the network is designed to use high-level variables describing the event an approach of building a direct dense neural network without additional complications was chosen. A rough hyperparameter search over the number of neurons, depth of the network, learning rate and activation function to be used was performed and the network was chosen based on those results. The resulting classifier contains an input sanitizer layer followed by four dense layers of decreasing width with swish activated neurons interlaced with batch normalization layers. The last layer uses a single sigmoid activated neuron to produce predictions between zero and one based on whether the input seems more like background or signal respectively. A schematic of the network structure is presented in Figure 20.3.

The relatively small amount of need for regularisation in the network is partially explained by the large amount of training samples but it also seems like the augmented training loss that includes the distance correlation term regularizes the training and prevents overfitting. The initial learning rate is set to lr$= 3 \cdot 10^{-4}$ and a learning rate reduction to one tenth of the previous rate is applied when the decrease in validation loss plateaus (does not produce a new lowest score) for 70 epochs during training. Training is run for 250 epochs at the end of which the network seems to have converged based on the training and validation loss functions. The relevant hyperparameters for the training are collected in Table 20.2.

The plots for distortion of background event m_{T} distribution in the separate test dataset and the DNN output distribution are shown in Figure 20.4 for the final training. Compared to the results shown in Figure 20.1 the amount of distortion on the background event distribution seems even less significant and this improvement can be explained with the addition of all the signal mass points to the training set in the full training. This provides natural planing to the signal samples as instead of being clustered around a few regions of m_{T} there are signal samples distributed more evenly across the spectrum.

In Figure 20.5 the loss and AUC values for the training and validation datasets are shown as

Training parameter	Value
Learning rate	$3 \cdot 10^{-4}$
λ	15
Learning rate schedule	Reduce after 70 epoch plateau
Minibatch size N	1024
Min-max scaling	Range $[-1, 1]$
Optimizer	Adam

Table 20.2: Relevant parameters for the training of the neural network classifier.

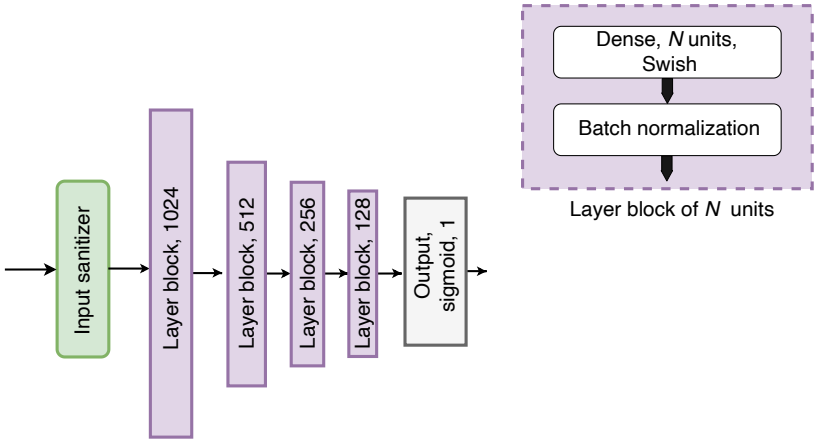


Figure 20.3: Network architecture of the classifier used for evaluating charged Higgs boson candidate events.

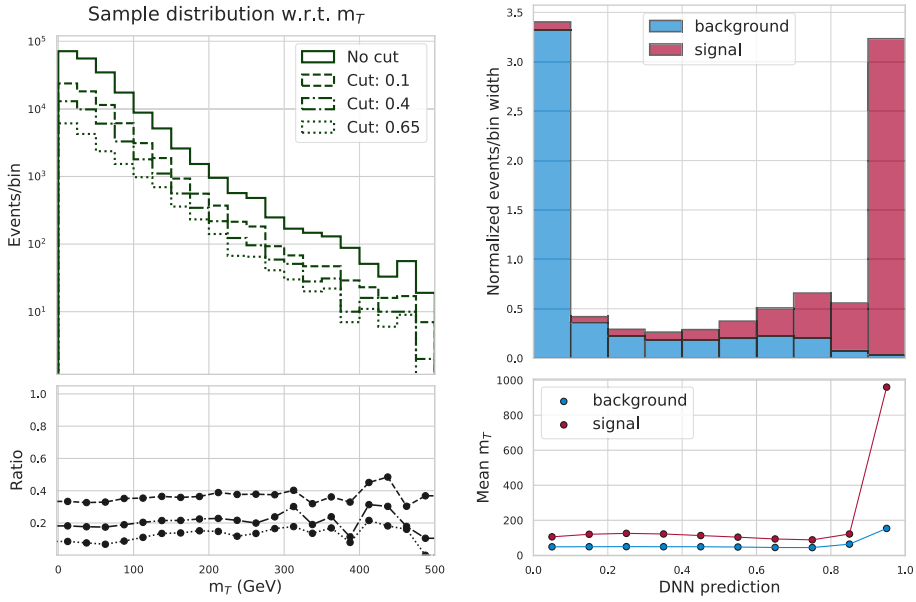


Figure 20.4: **Left**: The distortion of background sample m_{T} distribution at different cut values of the DNN output. **Right**: The DNN output value distribution for signal and background events. The mean m_{T} per bin is shown in the lower portion and can be seen to remain near constant for the background events indicating a level decorrelation between the DNN output and m_{T}.

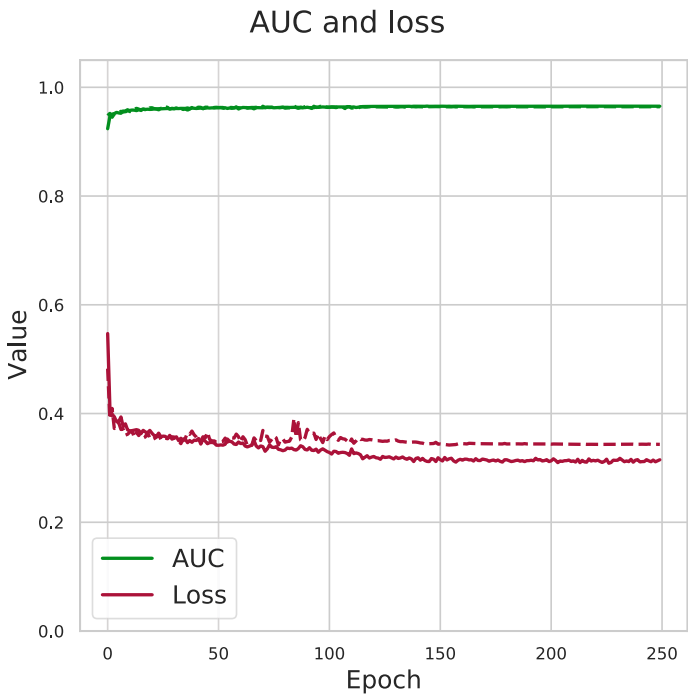


Figure 20.5: The training loss and AUC for the full classifier (trained on even samples). The continuous line displays the values measured on training set and the dashed line shows the values on the validation set.

a function of epochs.

20.7 Uncertainties introduced by the classifier

In itself the neural network is just a function that will output a reproducible value for given input, there is no inherent uncertainty in computing the output values for each event. Naturally the input variables given to the neural network contain their own uncertainties which are propagated through the selections including the neural network classifier by the means of variations.

Chapter 21

Background estimation

In order to determine a new particle has been found the background from known Standard Model processes has to be understood correctly, since signal will show up as a small excess of events passing the selections on top of the background processes. In the search targeting the final state with a hadronically decaying tau and jets, the largest backgrounds come from $t\bar{t}$ and QCD multijet production. Other smaller contributions result from single top quark production and electroweak events that include $Z/\gamma*$ events, W-boson production with associated jets and diboson production.

This analysis separates the backgrounds into three types based on the τ_h candidate in the event: jet$\rightarrow \tau_h$ events, $e/\mu \rightarrow \tau_h$ events and genuine-tau events. The QCD multijet background is estimated using a data-driven approach called the *fake factor method* [227] presented in Section 21.1. QCD multijet events are the dominant contribution to the jets$\rightarrow \tau_h$ background. Both the genuine-tau and $e/\mu \rightarrow \tau_h$ backgrounds are determined from simulation described in Sections 21.3 and 21.2 respectively.

While using different approaches measuring different backgrounds complicates the analysis, the end goal is to minimize the uncertainties in the final analysis result and as such improve the sensitivity of the analysis. Some processes are restricted by theoretical uncertainties like is the case with the QCD multijet production which warrants the use data-driven approach while others might be difficult to measure directly from data due to irreducible backgrounds so that one must rely on simulations.

21.1 Data-driven measurement of jets$\rightarrow \tau_h$ background

In order to determine the amount of jet$\rightarrow \tau_h$ events in the signal region based on data, an indirect approach is needed where the number of such events is determined in a control region without possible signal contamination and suitable *transfer factors* are calculated which can then be used to scale the number of events measured in the control region to give an estimate of the background in the signal region. This approach is called the *ABCD method* based on the different regions of the phase space used for the computation as is shown in Figure 21.1.

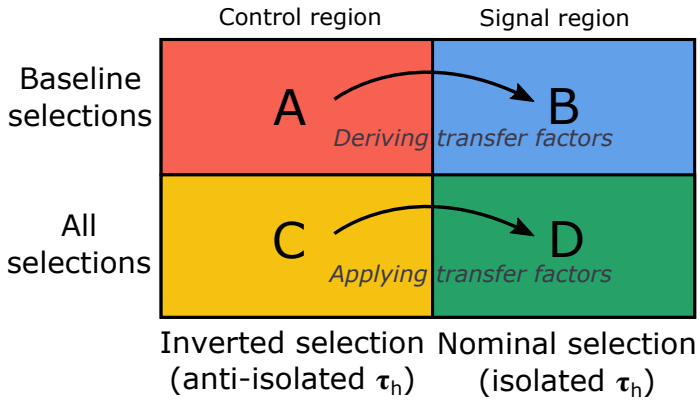


Figure 21.1: A schematic of the ABCD method. A baseline selection is used in the control region (A) to determine the transfer factors that scale the number of background events to the signal region (B). The transfer factors are then applied to scale the events passing all selections in the control region (C) in order to describe the background in the signal region after all selections (D).

21.1.1 Control region selection

A control region is chosen by using the events that do not pass the isolation condition in the τ_h selection. The *inverted selection* splits the dataset into two orthogonal sets where both of them contain at least the basic topology corresponding to the $H^+ \to \tau^+\bar{\nu}$ hadronic decay channel. The control region is enriched with jets$\to \tau_h$ candidates and it is dominated by QCD multijet events ($\approx 80\%$) with a contribution from electroweak and top events ($\approx 20\%$). There is still the trigger level isolation condition that even the control region events must have passed, so they can be considered as very loosely isolated τ_h candidates. As the isolation condition is inverted for determining this background, the estimate does not account for isolated e/μ that get misidentified as τ_h.

Roughly one fifth of this background that results from the top and electroweak events containing genuine τ_h candidates or isolated leptons that are misidentified as τ_h candidates is estimated from simulation that gets processed through the same inverted selection and is normalized to the theoretical cross sections. This contribution is then subtracted from the control sample with the goal of leaving only the QCD multijet contribution into the control region C:

$$N_{\mathbf{C},i}^{\text{jet}\to\tau_h} = N_{\mathbf{C},i}^{\text{data}} - N_{\mathbf{C},i}^{\tau\to\tau_h} - N_{\mathbf{C},i}^{e/\mu\to\tau_h} \tag{21.1}$$

The index i indicates that this subtraction is done in bins of p_T (< 60, 60-80, 80-100 and

$>$ 100 GeV) and $|\eta|$ ($<$ 0.6, 0.6-1.4 and $>$ 1.4) of the τ_{h} candidate. Binning in this 2D grid accounts for the correlation between τ_{h} candidate's p_{T} and $\vec{p}_{\mathrm{T}}^{\,\mathrm{miss}}$ and the geometrical dependency in the detector response with respect to η.

21.1.2 Normalization of the background measurement

In order to estimate the background in the signal region D based on the data in the control region C, transfer factors are determined after the baseline selections using regions A and B in Figure 21.1. This is done before the more refined selections that aim to drastically reduce the backgrounds as at this stage there is still sufficient number of events to keep statistical uncertainty related to the estimation in check and the possible signal is not yet observable above the background so it will not get absorbed into the background estimate.

The different processes contributing to the measurement in region A have a differing distribution of quark and gluon jets [227], which requires the use of different transfer factors for the QCD multijet and the electroweak/top components of the background. To account for this the transfer factors are estimated separately for the two contributions and then combined together as a weighted average.

For QCD multijet events we define the transfer factor as

$$R_i^{\mathrm{QCD}} \equiv \frac{N_{\mathrm{B},i}^{\mathrm{QCD}}}{N_{\mathrm{A},i}^{\mathrm{QCD}}}, \tag{21.2}$$

with $N_{\mathrm{B},i}^{\mathrm{QCD}}$ being the number of QCD multijet events that pass the nominal baseline selections and $N_{\mathrm{A},i}^{\mathrm{QCD}}$ the same for the inverted baseline selections. Similarly as for the control region C, the QCD multijet contribution in control region A is estimated from data by removing the contributions from electroweak/top events with genuine or fake τ_{h} candidate passing the selection using simulation. This measurement in control region A is then used to produce fit templates of the fraction of QCD multijet events as a function of $\vec{p}_{\mathrm{T}}^{\,\mathrm{miss}}$ using binned maximum likelihood fit. Under the assumption that the $\vec{p}_{\mathrm{T}}^{\,\mathrm{miss}}$ shape of the QCD multijet events is the same for both the inverted and nominal selection these templates will describe the expected fractions of QCD multijet events in background in the region B.

The description of the $\vec{p}_{\mathrm{T}}^{\,\mathrm{miss}}$ distribution for the QCD multijet background is achieved by using a combined Rayleigh, Gaussian and exponential function for the fit

$$f(x) = \frac{x-\mu_1}{\sigma_1^2} e^{\frac{-(x-\mu_1)^2}{2\sigma_1^2}} + \frac{1}{\sqrt{2\pi}\sigma_2} e^{-(x-\mu_2)^2} + \mu_3 e^{-\sigma_3 x}, \tag{21.3}$$

where μ_i and σ_i are the fit parameters used.

The electroweak/top component of the background is estimated directly from simulation by counting the events passing the nominal (inverted) selections that do not contain a genuine tau. Similarly as above, a template fit is produced using a combined Gaussian and exponential function fit

$$g(x) = \frac{1}{\sqrt{2\pi}\sigma_1} e^{-(x-\mu_1)^2} + \mu_2 e^{-\sigma_2 x}, \tag{21.4}$$

where μ_i and σ_i are the fit parameters. The transfer factors are defined as

$$R_i^{\text{EWK+top}} \equiv \frac{N_{\text{B},i}^{\text{EWK+top}}}{N_{\text{A},i}^{\text{EWK+top}}} \tag{21.5}$$

As the background measurement was done in bins i with respect to both p_{T} and $|\eta|$ of the τ_{h} candidate the normalization is done using the same binning, separately both components of the jet$\rightarrow \tau_{\text{h}}$ background. The template fits for both components are demonstrated in Figure 21.2.

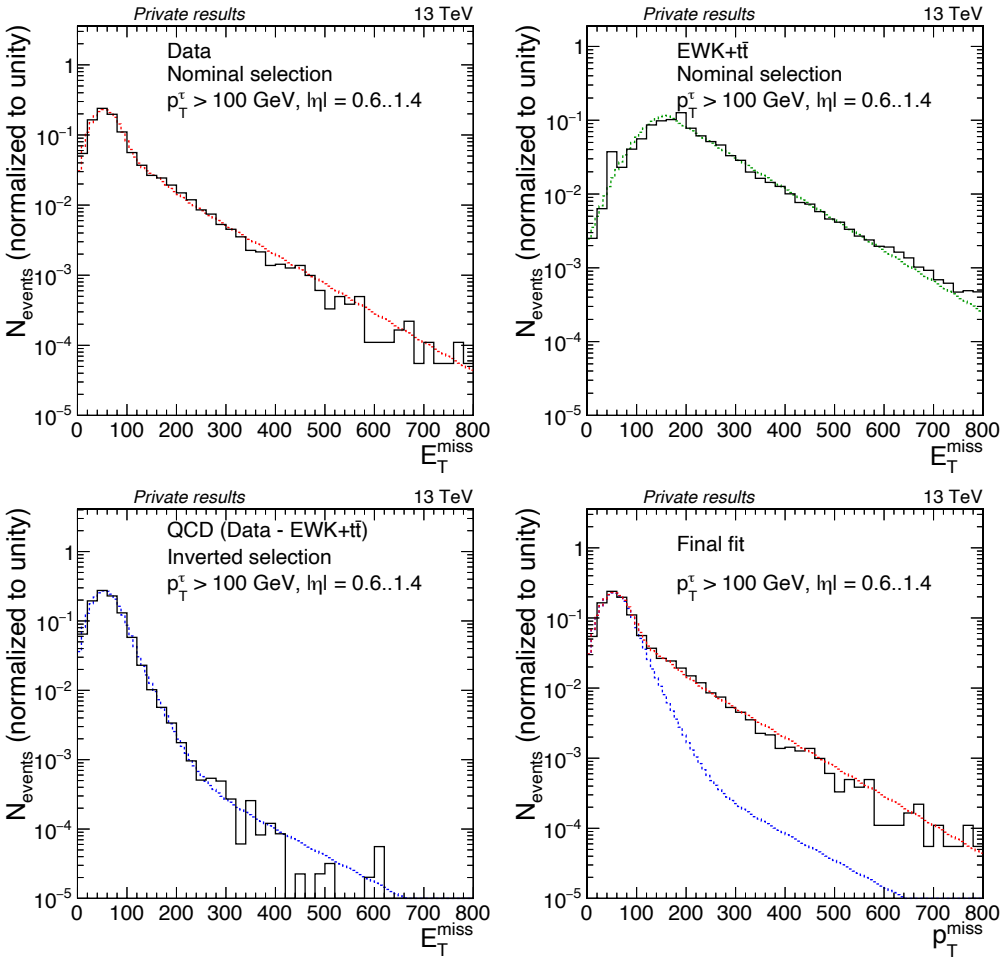


Figure 21.2: The fitted shapes of the $p_{\mathrm{T}}^{\mathrm{miss}}$ distribution. **Top left**: Data after the nominal baseline selection. **Top right**: Electroweak and top events after the nominal baseline selections. **Bottom left**: QCD multijet events after the inverted baseline selections. **Bottom right**: The final fit describing the jet$\rightarrow \tau_{\mathrm{h}}$ background in the signal region performed using the templates taken from **top right** and **bottom left** to fit the data **top left**, where the contribution from QCD multijets is shown separately in blue. This process of computing and fitting the templates is done separately in multiple bins of $|\eta|$ and p_{T}^{τ}, here shown are the results for $p_{\mathrm{T}}^{\tau} > 100$ GeV, $|\eta| = [0.6, 1.4]$

Since the selections after the baseline selections contain cuts that are designed to specifically suppress the QCD multijet component in the signal region, the relative fractions of QCD and electroweak/top events in the jet$\rightarrow \tau_{\mathrm{h}}$ are expected to be different after all selections compared to what they are after the baseline selections. In order to account for this, the weighted average of the transfer factors for the different components is done using the relative fraction for the backgrounds after all selections. This means that the final transfer factor in bin i for the jet$\rightarrow \tau_{\mathrm{h}}$ background is defined as

$$R_i \equiv w_i R_i^{\mathrm{QCD}} + (1 - w_i) R_i^{\mathrm{EWK+top}}, \tag{21.6}$$

where w_i is the fraction of QCD multijet events over all jet$\rightarrow \tau_{\mathrm{h}}$ events in the final event yield in the control sample after all selections:

$$w_i = \frac{N_{\mathrm{C},i}^{\mathrm{QCD}}}{N_{\mathrm{C},i}^{\mathrm{jet}\rightarrow\tau_{\mathrm{h}}}} = \frac{N_{\mathrm{C},i}^{\mathrm{data}} - N_{\mathrm{C},i}^{\mathrm{EWK+top}}}{N_{\mathbf{C},i}^{\mathrm{data}} - N_{\mathbf{C},i}^{\tau\rightarrow\tau_{\mathrm{h}}} - N_{\mathbf{C},i}^{e/\mu\rightarrow\tau_{\mathrm{h}}}} \tag{21.7}$$

This gives us the number of expected jet$\rightarrow \tau_{\mathrm{h}}$ events in the signal region D based on the data-driven estimation as

$$N^{\mathrm{jet}\rightarrow\tau_{\mathrm{h}}} = \sum_i \left(N_{\mathbf{C},i}^{\mathrm{data}} - N_{\mathbf{C},i}^{\tau\rightarrow\tau_{\mathrm{h}}} - N_{\mathbf{C},i}^{e/\mu\rightarrow\tau_{\mathrm{h}}} \right) R_i. \tag{21.8}$$

21.1.3 Systematic uncertainties

This data-driven background measurement has three different types of systematic uncertainties associated with it:

- Uncertainties from simulated genuine-tau and $e/\mu \rightarrow \tau_{\mathrm{h}}$ EWK+top events
- Limited precision of the transfer factors due to statistical uncertainties
- Statistical fluctuations in m_{T} shape for the jet$\rightarrow \tau_{\mathrm{h}}$ events in signal and control regions.

The first group of uncertainties resulting from the simulation of genuine-tau and $e/\mu \rightarrow \tau_{\mathrm{h}}$ events include uncertainties related to theoretical cross sections of the processes, identification of physics objects in data and the behaviour of the trigger selecting the events. These are propagated through the event selection and scaled down by the fraction of the simulated events in the control region. Since these simulated events are subtracted from the data

in the background estimation, they are treated as *anti-correlated* i.e. when all simulated events are varied upwards by 2.5% during estimation of final shape systematics, this variation is applied as a downwards variation to these simulated events that are being subtracted, in order propagate the variation in the correct direction through the jet$\rightarrow \tau_{\mathrm{h}}$ background estimation.

The statistical uncertainties limiting the precision of the transfer factors is calculated by propagating the statistical uncertainties for R_i^{QCD}, $R_i^{\mathrm{EWK+top}}$ and w_i through Eq. 21.1.2 defining the transfer factor R_i for the ith bin. The total uncertainty from the normalization process is then calculated as an uncorrelated sum of the R_i uncertainties since they are statistical in origin.

The shapes of the m_{T} distributions of the control and signal regions are obtained after all selections and normalized to unity. The bin-by-bin uncertainty due to limited statistics are computed for both distributions, and then the normalized signal region distribution is divided by the normalized control region distribution. The uncertainty of the resulting distribution is calculated by propagating the errors in the quotient and applying this as a shape uncertainty to the final m_{T} distribution.

21.1.4 Validation of the data-driven measurement

Different assumptions of the method are validated separately. First the decision to derive the transfer factors early on in the selections in order to have sufficient statistics for a reliable estimate and avoid absorbing possible signal to the data-driven fit for the background is validated. This is done by deriving another set of transfer factors after an additional selection step, the b-jet selection, is applied and the two sets of transfer factors are compared together. The factors are found to be compatible within statistical uncertainties, so the chosen approach of computing the factors after baseline selections is considered acceptable.

The method assumes the m_{T} shape to be compatible between the control region C and the signal region D, since only the overall number of events is corrected in the τ_{h} bins by the normalization factor and not the shape. To study the validity of this assumption an additional validation region is chosen where the b-jet selection condition is inverted in order to ensure there is no significant signal contamination. In this region the m_{T} distributions of events after all selections that pass the τ_{h} isolation condition are compared with the events failing the isolation condition i.e. the nominal and inverted selections. The genuine-tau and $e/\mu \rightarrow \tau_{\mathrm{h}}$ contributions are subtracted from the data in order to compare only the QCD multijet events. This comparison is shown in Figure 21.3 for both R_τ categories and the shapes are found compatible within statistical uncertainties.

Finally the chosen approach to perform the measurement in bins of the τ_{h} candidate $p_{\mathrm{T}}^{\mathrm{miss}}$ and $|\eta|$ is validated by comparing the results from achieved with the chosen binning to alternative binnings and an inclusive binning. The differing binning approaches are found to agree within uncertainties.

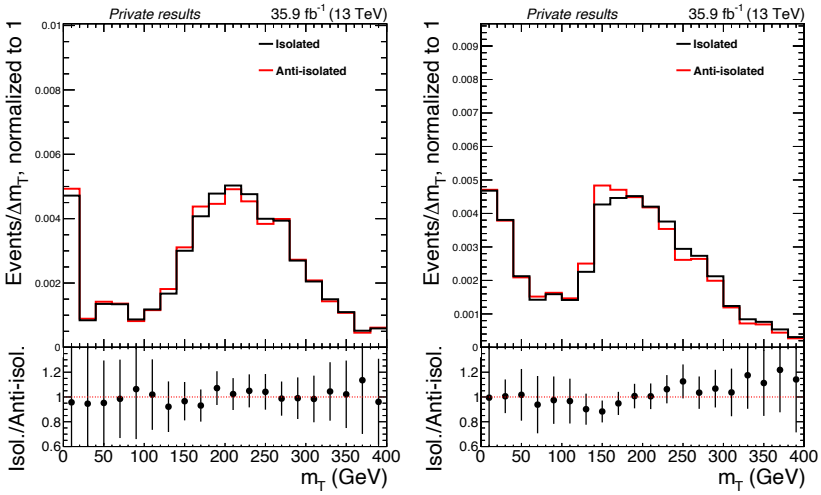


Figure 21.3: Closure test for the m_{T} distributions in the validation region for the jet$\rightarrow \tau_{\mathrm{h}}$ background. Statistical uncertainties are displayed in the errorbars of the ratio plot. The distribution for events passing the τ_{h} isolation requirement are found to be compatible with the distribution for the events failing it in both categories of the R_{τ} variable. **Left**: $R_{\tau} \leq 0.75$. **Right**: $R_{\tau} > 0.75$.

21.2 Estimation of $e/\mu \rightarrow \tau_{\mathrm{h}}$ background with simulation

Electroweak and top events where a lepton gets misidentified as τ_{h} are estimated with simulated samples. If the reconstructed τ_{h} candidate is matched to a generator-level electron or muon within ΔR of 0.1, the event is considered as $e/\mu \rightarrow \tau_{\mathrm{h}}$ background.

The largest contribution to this background comes from $\mathrm{t\bar{t}}$ events. Out of the $\mathrm{t\bar{t}}$ events passing the selections up to τ_{h} selection, roughly 4% contain a misidentified τ_{h} candidate from a lepton.

21.3 Estimation of genuine-tau background with simulation

Genuine-tau background is identified as the simulated events where the τ_{h} candidate is associated with a generator-level tau within $\Delta R \leq 0.1$ of the reconstructed τ_{h}. The contributions for the genuine-tau background come from $\mathrm{t\bar{t}}$, single top, W+jets, Z/γ and diboson processes, out of which the $\mathrm{t\bar{t}}$ is the dominant component.

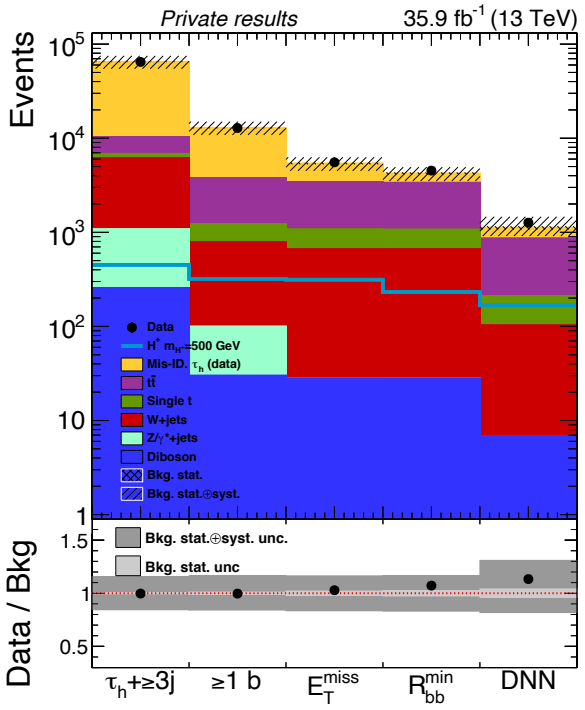


Figure 21.4: Composition of simulated events passing each selection step. Dots represent the number of events in real data passing the selection and the bins contain the normalized simulated background events passing the selection. The shaded areas represent the total uncertainty in the bin.

21.4 Background selection efficiencies

The composition of events based on the simulated samples passing each selection step is presented in Figure 21.4. Additionally the number of passing events in data is shown by dots. The shaded areas reflect the total systematic and statistical uncertainty in the bin. A signal sample simulated with $m_{\mathrm{H}^+} = 500$ GeV is displayed in the selection flow plot as well.

The selection flow plot demonstrates how different selections affect different backgrounds and signal events. Clearly the b-jet tagging and the selection based on it have a drastic effect in reducing the background events. Unfortunately it is also the single largest cut in the expected amount of signal events passing the selection. The event yields for genuine tau background events passing the different selections including the standard selections are shown in Table 21.1. A similar table for some of the signal mass points is presented in Table 21.2

The expected distributions of the main variables used in the selections and their agreement

Process	W+jets	$t\bar{t}$	Single top	Z/γ	Diboson
All events	3468584.2 ± 1309.0	438670.7 ± 252.6	1474102.0 ± 12713.6	171294.8 ± 254.7	
Trigger	1513363.0 ± 5559.6	1032208.9 ± 707.7	139259.2 ± 145.4	321856.0 ± 5942.6	59503.3 ± 129.5
Tau selection	13891.0 ± 534.9	9598.7 ± 67.7	1540.6 ± 15.6	3284.6 ± 601.7	784.8 ± 14.7
Trg. and tau id. SF	8582.1 ± 371.3	5759.9 ± 44.7	1013.6 ± 11.2	2100.7 ± 415.3	520.4 ± 10.4
Lepton veto	8419.0 ± 367.7	3996.7 ± 37.5	783.8 ± 9.9	1425.9 ± 346.4	366.2 ± 8.9
Jet selection	5085.9 ± 289.9	3487.4 ± 35.1	642.3 ± 9.0	820.1 ± 250.8	255.4 ± 7.5
B-jet selection	674.7 ± 105.0	2607.3 ± 30.3	447.1 ± 7.5	63.2 ± 63.2	29.6 ± 2.6
$\vec{p}_{\mathrm{T}}^{\,\mathrm{miss}}$ selection	630.2 ± 103.4	2352.9 ± 28.7	411.9 ± 7.2	0.0 ± 0.0	27.9 ± 2.5
Angular selection	630.2 ± 103.4	2286.4 ± 28.3	406.2 ± 7.2	0.0 ± 0.0	27.7 ± 2.5
DNN selection	95.4 ± 39.1	647.1 ± 15.2	106.6 ± 3.7	0.0 ± 0.0	6.8 ± 1.3

Table 21.1: Estimated event yields for the genuine tau backgrounds for each selection step. The numbers are normalized to the theoretical cross sections. The values correspond to the $R_\tau < 0.75$ category

Process	$H^{\pm}$ (120 GeV)	$H^{\pm}$ (200 GeV)	$H^{\pm}$ (500 GeV)	$H^{\pm}$ (2000 GeV)
All events	44438.2 ± 284.8	90680.5 ± 540.9	127194.0 ± 679.6	3365186.5 ± 3803.9
Trigger	14990.8 ± 166.0	46806.2 ± 384.6	107856.3 ± 622.0	3209784.0 ± 3706.1
Tau selection	631.0 ± 33.5	3206.4 ± 98.6	11912.7 ± 203.4	449882.9 ± 1360.4
Trg. and tau id. SF	409.8 ± 22.3	2227.8 ± 70.2	9780.2 ± 169.0	414249.1 ± 1254.7
Lepton veto	280.1 ± 18.7	1668.7 ± 61.1	7659.9 ± 148.8	312830.4 ± 1087.5
Jet selection	170.6 ± 15.4	1023.3 ± 51.7	4308.3 ± 123.2	288739.4 ± 1053.4
B-jet selection	133.3 ± 13.6	746.8 ± 43.8	3118.2 ± 102.5	185097.5 ± 841.0
$\vec{p}_{\mathrm{T}}^{\,\mathrm{miss}}$ selection	122.9 ± 12.7	670.1 ± 41.6	2999.4 ± 99.9	182973.0 ± 832.9
Angular selection	109.2 ± 12.3	644.2 ± 40.1	2223.5 ± 86.3	108361.1 ± 644.6
DNN selection	42.7 ± 7.5	333.1 ± 27.8	1621.7 ± 71.7	49521.0 ± 433.7

Table 21.2: Signal sample event yields for each selection steps. The samples are normalized to a production cross section of 1 pb. The values correspond to $R_\tau < 0.75$.

with data is displayed at the point of selection in Figure 21.5.

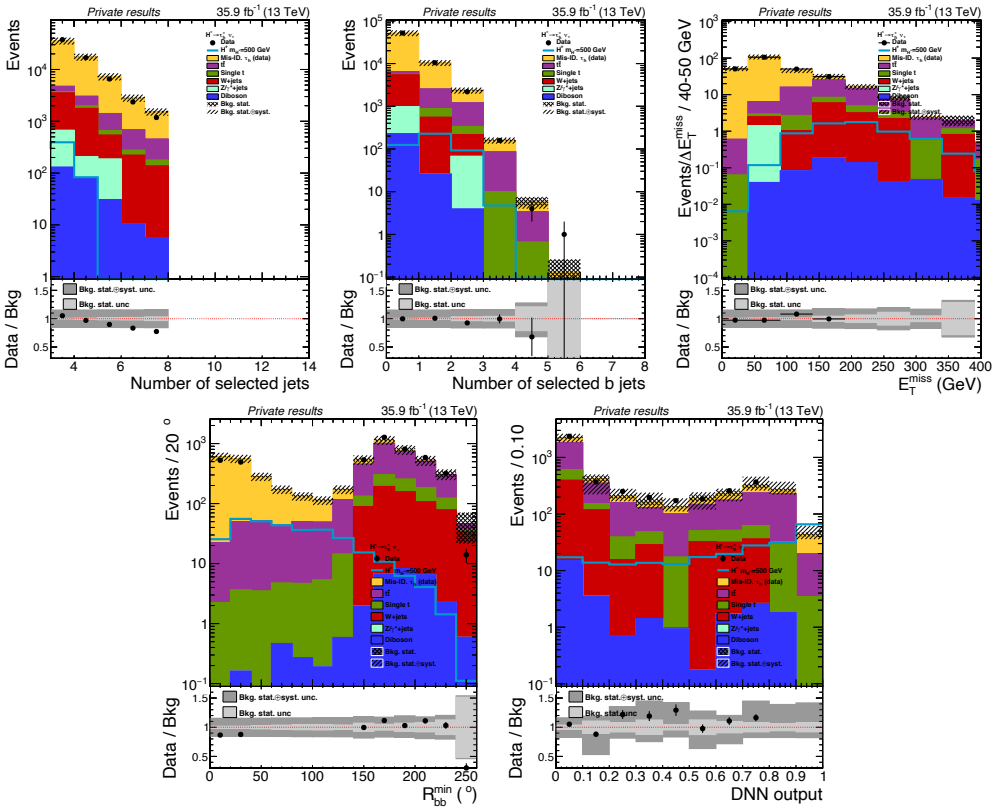


Figure 21.5: Distributions for the five main variables used for the selection cuts at the point of selection. for the $R_\tau < 0.75$ category. From top left: Number of identified jets in the event, number of b-jet candidates passing identification, missing transverse energy, the angle between the τ_h candidate and the nearest jet and the deep neural network predicted values.

Chapter 22

Systematic uncertainties and corrections

In an enormously complicated process, such as measuring the interactions between subatomic particles in collisions, there are imperfections in the methods used that are encapsulated into uncertainties which limit how confidently one can claim to have measured something. In order to capture and quantitatively describe uncertainties related to the detector apparatus and reconstruction methods used, intricate models describing the workings of different parts of the detector and their interactions with the particles being measured have been built and these models are used to compare the actual measurements with our theoretical understanding of the phenomena. Any discrepancies between the two are investigated and an effort is made to either fix the source of the discrepancy or assign a correction factor and an associated uncertainty to make the two match. The summary of the uncertainties is presented in Table 22.1 and the rest of this chapter details the various corrections and uncertainties that are accounted for in the analysis.

Source	shape	$H^{\pm}$	jet $\rightarrow \tau_h$	$t\bar{t}$	single top	electroweak
τ_h identification	Yes	2.7	<0.1	0.2	0.7	2.0
Trigger efficiency	Yes	2.8	0.5	2.6	2.6	3.7
Lepton veto efficiency	No	0.2	–	0.5	0.3	<0.1
b-jet identification	Yes	2.7	0.5	2.5	1.1	2.7
τ_h energy scale	Yes	3.2	0.5	3.4	0.1	5.8
Jet energy scale	Yes	5.7	0.3	1.5	4.7	4.1
Jet energy resolution	Yes	3.5	1.6	1.2	4.9	4.0
Unclust. $\vec{p}_T^{\,miss}$ ES	Yes	<0.1	<0.1	<0.1	<0.1	<0.1
Jet misid.as τ_h	Yes	–	6.3	–	–	–
Pileup	Yes	<0.1	<0.1	<0.1	<0.1	<0.1
Top mass	No	–	0.6	2.8	2.2	–
Acceptance (scale, PDF)	No	5.1	0.6	2.1	2.1	11.8
Cross section (scale, PDF)	No	–	0.9	4.8	2.2	9.4
Integrated luminosity	No	2.5	0.6	2.5	2.5	2.5
Total		10.5	6.7	8.4	8.4	18.0

Table 22.1: The systematic uncertainties and their effect on the event yield of the analysis for signal and different backrounds in % summed over both R_τ categories. Signal with $m_{H^{\pm}} = 500$ GeV is shown here. Shape uncertainties modify both the shape and normalization of the final m_T distribution while the other uncertainties affect only the normalization.

22.1 Selection efficiencies

22.1.1 Trigger efficiencies

The τ_h efficiency is measured in data using $Z/\gamma* \rightarrow \tau^+\tau^-$ events where one of the taus decay hadronically and the other into a muon. This type of events can be selected from the data with a high signal purity (> 90%).

For the τ_h trigger the efficiency simply means the fraction of these events where both the muon and the tau are noticed by the trigger. The muon serves as a "tag" giving the measurement of the p_T for the τ_h candidate so that the efficiency of finding and passing a genuine-tau event can be measured as a function of p_T. This method is called the tag-and-probe method [228] and it takes advantage of the excellent muon detection and measurement capabilities of the CMS detector.

The $\vec{p}_T^{\,miss,\,calo}$ trigger efficiency can be measured using a single τ_h trigger with no $\vec{p}_T^{\,miss}$ requirement and comparing that to a trigger with the same τ_h selection and a $\vec{p}_T^{\,miss}$ threshold. The efficiency is defined as the ratio of events passing the stricter trigger (signal trigger) over the events passing the looser selection (monitor trigger).

For both triggers there is a discrepancy between the measured and simulated efficiencies. This discrepancy is corrected by computing scale factors from the ratios of fitted efficiencies for the data ϵ_{data} and simulation ϵ_{sim}. These corrections are applied to the event yields of simulated samples as a function of the p_T of the τ_h candidate and $\vec{p}_T^{\,miss}$ of the event.

The uncertainties of the fits are propagated into the final m_{T} distribution as four independent nuisance parameters for both the τ_{h} and $\vec{p}_{\mathrm{T}}^{\,\mathrm{miss}}$ triggers by varying the efficiencies up and down by the fit uncertainties producing the combinations $\epsilon_{\mathrm{data}}/\epsilon_{\mathrm{sim}}^{\mathrm{up}}$, $\epsilon_{\mathrm{data}}/\epsilon_{\mathrm{sim}}^{\mathrm{down}}$, $\epsilon_{\mathrm{data}}^{\mathrm{up}}/\epsilon_{\mathrm{sim}}$ and $\epsilon_{\mathrm{data}}^{\mathrm{down}}/\epsilon_{\mathrm{sim}}$.

22.1.2 τ_{h} isolation and identification

Detailed description of the τ_{h} identification efficiency measurement is presented in [219]. The study is done using the tag-and-probe method to compare data and simulation and an agreement within statistical uncertainties is obtained. The small difference in the mean values is corrected by applying a scalar scale factor of 0.99 to all simulated events with a τ_{h} candidate from a genuine tau lepton. A 5% uncertainty from the measurement is applied as a normalization uncertainty to this analysis.

The effect of background events with isolated electrons or muons that get misidentified as τ_{h} discussed in Section 21.2 is corrected with scale factors in the range of 1.40 ± 0.12 to 1.90 ± 0.30 (1.12 ± 0.04 to 2.39 ± 0.16) for electrons (muons) where the values and uncertainties grows towards higher $|\eta|$. These are propagated and included as independent shape uncertainties for electrons and muons misidentified as τ_{h} in the final m_{T} distribution.

An additional uncertainty for the τ_{h} candidates with a large p_{T} value is applied as a shape uncertainty. This concerns the events with $p_{\mathrm{T}} > 200$ GeV where an uncertainty of $^{+5}_{-35}\,\%$ $p_{\mathrm{T}}/\mathrm{TeV}$ is used to account for the lack of data in measuring the efficiency of the τ_{h} identification, requiring the use of a polynomial fit to extrapolate the efficiency measurement to this region.

22.1.3 B-jet identification

There is a discrepancy between the b-jet identification performance between simulation and data. This is corrected by including scale factors to the simulated events [229]. This correction distorts the shapes of the m_{T} distribution so additional uncertainties are included to account for this effect.

Each simulated event is assigned a *per-event scale factor* for the correction which takes into account the *per-jet scale factors* determined as a function of p_{T}, $|\eta|$ and jet flavour for each jet in the event. Jet flavours are denominated as g, b, c and u/d/s where g refers to gluons and the other letters refer to the quark that originated the jet based on the simulation ground truth. In this analysis it has been determined that the $|\eta|$ dependency for the corrections is negligible. Additionally c-jets are treated as b-jets for the purposes of computing the corrections.

The scale factors quantifying the difference of b-jet identification efficiency between data and simulation are determined as

$$f_{\text{tag}}(p_{\text{T}}) = \frac{\epsilon_{\text{tag}}^{\text{data}}(p_{\text{T}})}{\epsilon_{\text{tag}}^{\text{simulation}}(p_{\text{T}})}, \quad f_{\text{tag}}(p_{\text{T}}) = \frac{\epsilon_{\text{mistag}}^{\text{data}}(p_{\text{T}})}{\epsilon_{\text{mistag}}^{\text{simulation}}(p_{\text{T}})}, \tag{22.1}$$

where ϵ_{tag} (ϵ_{mistag}) is the b-jet (mis)tagging efficiency. The efficiencies are determined from a simulated sample of $\text{t}\bar{\text{t}}$ events after the baseline selections described in 19.3.

The probability for any single jet to pass the b-jet identification step of the analysis selections is independent of the other jets in the event, so the probability P for an event to pass the identification step is computed as

$$P = \prod_{i=1}^{N_{\text{b,c tagged}}} \epsilon_{\text{tag},i} \prod_{j=1}^{N_{\text{b,c not tagged}}} (1 - \epsilon_{\text{tag},j}) \times \prod_{k=1}^{N_{\text{uds,g tagged}}} \epsilon_{\text{mistag},k} \prod_{l=1}^{N_{\text{uds,g not tagged}}} (1 - \epsilon_{\text{mistag},l}). \tag{22.2}$$

The per-event scale factor needed corrects for the discrepancy between the data and simulated events passing this selection:

$$\text{SF} = \frac{P(\text{data})}{P(\text{simulation})}, \tag{22.3}$$

which can be rewritten using Equations 22.1.3 and 22.1.3 in terms of per-jet scale factors ϵ and b-jet (mis)tagging efficiencies f as:

$$\text{SF} = \prod_{i=1}^{N_{\text{b,c tagged}}} f_{\text{tag},i} \prod_{j=1}^{N_{\text{b,c not tagged}}} \left(\frac{\frac{1-f_{\text{tag},j}}{\epsilon_{\text{b,c},j}}}{1 - \epsilon_{\text{b,c},j}} \right) \tag{22.4}$$

$$\times \prod_{k=1}^{N_{\text{uds,g tagged}}} f_{\text{mistag, k}} \prod_{l=1}^{N_{\text{uds,g not tagged}}} \left(\frac{\frac{1-f_{\text{mistag},l}}{\epsilon_{\text{uds,g},l}}}{1 - \epsilon_{\text{uds,g},l}} \right). \tag{22.5}$$

The uncertainties accompanying the use of these scale factors are computed as per-event uncertainties that include the contributions from the per-jet scale factors and the measured b-jet (mis)tagging efficiencies. The uncertainties of the per-jet scale factors f_{tag} and f_{mistag}

and the uncertainties in the measured efficiencies ϵ are assumed uncorrelated and they are propagated through Equation 22.1.3. The uncertainties assigned to c-jets are conservatively taken as being twice the b-jet uncertainties. The uncertainties of b-jet tagging and mistagging are treated in the final analysis as two independent shape nuisances as they are assumed uncorrelated.

22.1.4 Lepton isolation and identification

The uncertainty in the veto of isolated electrons and muons is defined as

$$E = \frac{N^{\text{vetoed}}}{N^{\text{selected}}} \times \Delta_{\text{Id.}}, \tag{22.6}$$

where N^{vetoed} is the number of events removed by the veto, N^{selected} is the number of events passing the veto step and Δ_{Id} is the uncertainty in the identification and isolation efficiencies. $\Delta_{\text{Id.}}$ is 1% for electrons and 2% for muons. The resulting uncertainties are applied as two scalar normalization uncertainties for electrons and muons independently.

22.2 Energy scales

The reconstructed jets, τ_{h} candidates and the missing energy $\vec{p}_{\text{T}}^{\text{miss}}$ have associated systematic uncertainties in their energy measurements and the methods that were used to calibrate them. These uncertainties are included in the analysis as shape uncertainties that are produced by varying the energy scales up and down and rerunning the analysis workflow on the events. This results in up and down varied m_{T} distributions that can be used to define the shape uncertainties.

22.2.1 τ_{h} energy scale

The energy measurements of the τ_{h} candidates are corrected by using $\text{Z}/\gamma* \to \tau^{+}\tau^{-}$ events from data and simulation with $e\tau_{\text{h}}$ and $\mu\tau_{\text{h}}$ final states, where the other lepton provides a more accurate energy measurement that can be compared to the energy measured for the reconstructed τ_{h} candidate and the known invariant mass of the $l\tau_{\text{h}}$ system in these events [219].

The corrections based on this method vary between 0.995 and 1.011 and they are applied to τ_{h} candidates with energies up to 400 GeV and the associated uncertainty of the correction is determined as $\pm 1.2\%$. For τ_{h} candidates with energies above the threshold no correction is applied but an additional systematic uncertainty of 3% is assigned.

22.2.2 Jet energy scale corrections and resolution

The uncertainties associated with the multi-step process of applying jet energy corrections is estimated by propagating all of the associated uncertainties of the jet energy measurement through the calibration workflow and accounting for possible correlations. The process is documented in [168]. This results in a function describing the jet uncertainties as a function of jet p_{T} and η that can be used to vary jet energy measurements up and down to produce shape uncertainties out of the final m_{T} distribution.

Additional uncertainties are included due to the jet energy resolution mismatch between data and simulated events. The energies of simulated jets are reconstructed more accurately than those from data and this mismatch is corrected by including a p_{T} dependent smearing factor to the four-momenta of simulated jets [168]. The uncertainty associated with the jet energy resolution is propagated to the final m_{T} distribution through up and down variations and shape uncertainties are derived from it.

22.2.3 $\vec{p}_{\mathrm{T}}^{\,\mathrm{miss}}$ energy scale

Since the changes in the jet energies and resolution affect the sum four-momenta measured in the event, they need to be propagated to the type-I corrected $\vec{p}_{\mathrm{T}}^{\,\mathrm{miss}}$ measurement as well. Additional uncertainties are included from the unclustered energy in the detector after reconstruction of the physics objects. It is evaluated from the measured momentum resolutions of PF candidates, and the most significant contributions are from neutral hadrons in the HCAL and the particles reconstructed from the HF measurements [230].

22.3 Jet$\rightarrow \tau_{\mathrm{h}}$ background estimation

The method for performing the jet$\rightarrow \tau_{\mathrm{h}}$ measurement and estimating the associated uncertainties is described in detail in Section 21.1. The resulting shape uncertainty is up to 6.3%.

22.4 Cross section uncertainties

Computations for the process cross sections in the proton-proton collisions have their own set of uncertainties resulting from the choice of renormalization and factorization scales and the uncertainties in the parton distribution functions [231]. In case the process involves a top quark i.e. the $\mathrm{t\bar{t}}$ and single top quark events additionally an uncertainty related to the top quark mass m_{t} is included. This uncertainty is estimated by varying m_{t} value by 1.0 GeV around the nominal value 172.5 GeV. The uncertainties are included as normalization uncertainties for each background process.

The total cross section uncertainty for the dominant $\mathrm{t\bar{t}}$ background is $^{+4.7}_{-5.5}$ at $\sqrt{s} = 13$ TeV.

22.5 Acceptance uncertainties

Additionally the uncertainties of the chosen renormalization and factorization scales and the parton distribution function affect the selection efficiency of simulated events. These uncertainties are referred to as *acceptance uncertainties*.

The PDF acceptance uncertainty is estimated by simulating 100 replicas for each sample from the PDF probability distribution and using the standard deviation of the replicas as a normalization uncertainty. For $\mathrm{t\bar{t}}$ and single top backgrounds the uncertainty is $^{+0.3}_{-2.0}$ and $^{+4.6}_{-3.3}$ for the other backgrounds. For signal samples the uncertainty is determined to be $^{+1.7}_{-0.4}$.

The effect of renormalization and factorization uncertainties are determined by varying the two scales up and down by factors of 0.5 and 2.0, and running the analysis workflow on the events to determine the effect on the final m_{T} distribution. The shapes of the resulting distributions do not differ from the nominal distribution's shape, so the uncertainties are applied as normalization uncertainties. The uncertainty on $\mathrm{t\bar{t}}$ and single top backgrounds are 2.0% and 5.0% for the other backgrounds. For signal samples with up to $m_{\mathrm{H}^{\pm}} = 750$ GeV the uncertainty is 4.8% and for masses above that it is 1.2%.

22.6 Pileup modeling

In simulated events the pileup is sampled from a predefined distribution of events. As the actual pileup in the collisions dependes on the instantaneous luminosity at each bunch crossing in the detector, the simulated pileup is reweighted to make the distribution match the measured pileup distribution in data.

Uncertainty of this reweighting process is estimated by varying the inelastic pp cross section by $\pm 5\%$ around the nominal value and performing the reweighting again using the up and down varied cross section. These variations are propagated to the m_{T} distribution as shape uncertainties.

22.7 Signal modeling

The $\mathrm{H}^{\pm}$ signal samples are affected by the same uncertainties applied to the other simulated samples. For the light $\mathrm{H}^{\pm}$ samples with $m_{\mathrm{H}} \leq 165$ GeV the top quark related uncertainties are also included (and fully correlated with the $\mathrm{t\bar{t}}$ uncertainties), since the dominant production channel is assumed to be through decay of a top quark.

Intermediate-mass samples ($m_{\mathrm{H}} = [165, 175]$ GeV) are corrected for an excessively high selection efficiency which results from the use of leading order (LO) simulation samples in that region instead of next to leading order (NLO) samples used elsewhere analysis.

A systematic uncertainty is associated with this correction based on statistical uncertainties in ratios of NLO and LO events in the intermediate mass region used to calculate the correction and it is used as a normalization uncertainty. The treatment of the intermediate mass region is detailed in [203].

22.8 Luminosity measurement

The integrated luminosity is estimated to have an uncertainty of 2.5% [232]. This uncertainty is applied to all simulated background processes.

Chapter 23

Results

The dataset used here is the dataset collected by the CMS experiment during 2016 and the corresponding simulated datasets as described in Section 19.1. As the analysis for this data has already been presented at [203], the results here are not unblinded again. Instead the changes in the expected event counts and resulting expected limits are inspected with and without the use of deep neural network classifier.

23.1 Transverse mass distributions

After all selections the remaining events are binned into a histogram based on the reconstructed transverse mass between the τ_h candidate and $\vec{p}_\mathrm{T}^{\,\mathrm{miss}}$. The bin width is chosen so that sufficient statistics are contained in each bin causing the higher m_T range bins to be wider. The transverse mass histograms are produced separately for both R_τ categories. The blinded transverse mass distributions for the analysis with the deep neural network classifier and without it are shown in Figure 23.1. The deep neural network selection reduces the number of events by roughly 60% and by design it does so evenly across the m_T spectrum so the effect is subtle. This working point was chosen after testing different cut-off values and choosing the best performing one.

The portion of the mass spectrum where charged Higgs boson signal is thought possible has been kept blinded in this analysis and only the expected limits will be computed.

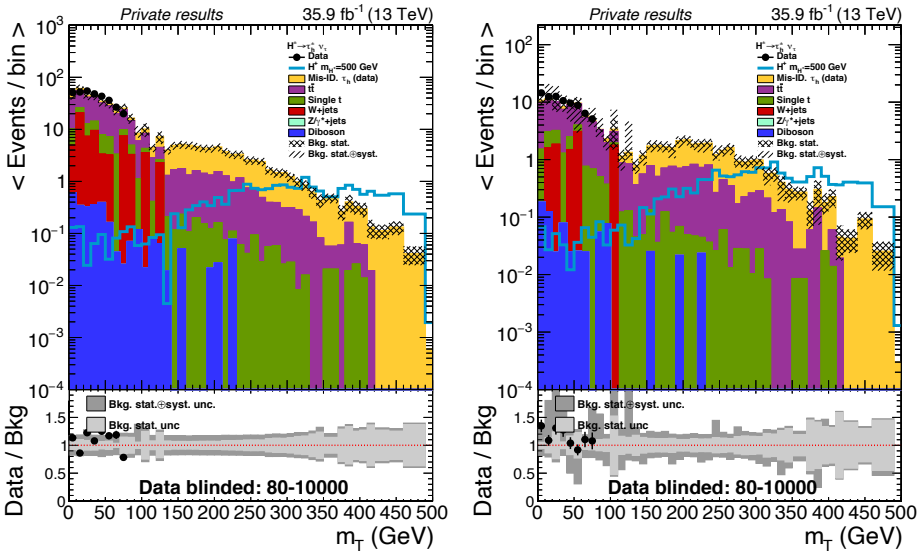


Figure 23.1: **Left:** Transverse mass distribution without the DNN selection. **Right:** Transverse mass distribution after the DNN selection.

23.2 Exclusion limits

Based on the approach outlined in Section 19.2 the expected exclusion limits are computed both with and without the deep neural network selection using the reconstructed transverse mass distribution as the summary statistic. Both R_τ categories are fitted simultaneously to find the 95% confidence level exclusion limit as they represent two orthogonal datasets providing independent measurements. The model independent limits are presented in Figure 23.2 for the $H^\pm \to \tau^\pm \nu_\tau$ in the hadronic tau decay channel.

Comparison of the limits shows that the effect of adding the last classification step with the deep neural network in this form has a negative effect on the overall sensitivity of the analysis. The results are also collected in Tables 23.1 and 23.2.

In Figure 23.3 the ratio of the limits over the limits produced by the original analysis are shown and the relative change caused by the new selection step can be read. The expected limits deteriorate between 3% and almost 400% with the new analysis step. While the classifier seems to remove more background than signal candidate events still the ratio of events removed in the cut is not beneficial enough.

This result can be interpreted in two ways. Either the classifier training has failed to find a useful solution to the problem and some other combination of hyperparameters and architecture might have resulted in a better classifier. Or the constraint of avoiding correlations with the reconstructed transverse mass combined with the chosen input variables did not contain enough discriminative information to produce a useful classifier.

$m_{H^\pm}$	Expected limit					Observed
(GeV)	-2σ	-1σ	median	$+1\sigma$	$+2\sigma$	limit
80	10.42664	13.83144	19.41250	27.53730	37.50197	Blinded
90	10.72902	14.24902	19.83125	27.81512	37.84097	Blinded
100	9.34210	12.60568	17.78125	25.50682	35.21391	Blinded
120	6.50043	8.73598	12.28125	17.51928	24.10220	Blinded
140	1.68782	2.26553	3.21250	4.58265	6.30460	Blinded
150	1.17228	1.58180	2.23125	3.16511	4.33871	Blinded
155	1.16383	1.56409	2.19883	3.11911	4.27567	Blinded
160	1.11090	1.49295	2.09883	2.97726	4.05461	Blinded
165	0.83646	1.11965	1.56875	2.23783	3.05892	Blinded
170	0.58057	0.78024	1.09687	1.56470	2.13881	Blinded
175	0.51390	0.69342	0.97812	1.39530	1.90725	Blinded
180	0.44822	0.59997	0.84062	1.19916	1.62854	Blinded
200	0.43832	0.58907	0.82812	1.17472	1.59981	Blinded
220	0.24411	0.32675	0.45781	0.65307	0.88692	Blinded
250	0.19456	0.26115	0.37031	0.52825	0.72675	Blinded
300	0.09974	0.13388	0.18984	0.27081	0.37257	Blinded
400	0.03892	0.05266	0.07520	0.10966	0.15475	Blinded
500	0.01537	0.02159	0.03213	0.04865	0.07068	Blinded
750	0.00258	0.00394	0.00674	0.01109	0.01733	Blinded
800	0.00186	0.00305	0.00518	0.00889	0.01397	Blinded
1000	0.00130	0.00209	0.00361	0.00655	0.01064	Blinded
1500	0.00058	0.00104	0.00205	0.00395	0.00616	Blinded
2000	0.00047	0.00084	0.00166	0.00320	0.00499	Blinded
2500	0.00046	0.00077	0.00146	0.00287	0.00436	Blinded
3000	0.00046	0.00077	0.00146	0.00287	0.00436	Blinded

Table 23.1: Expected 95% CL exclusion limits for $\sigma_{H^\pm}\mathcal{B}(H^\pm \to \tau^\pm\nu_\tau)$ for the baseline analysis. The $\pm 1(2)\sigma$ corresponds to the 1 (2) standard deviation from the median value.

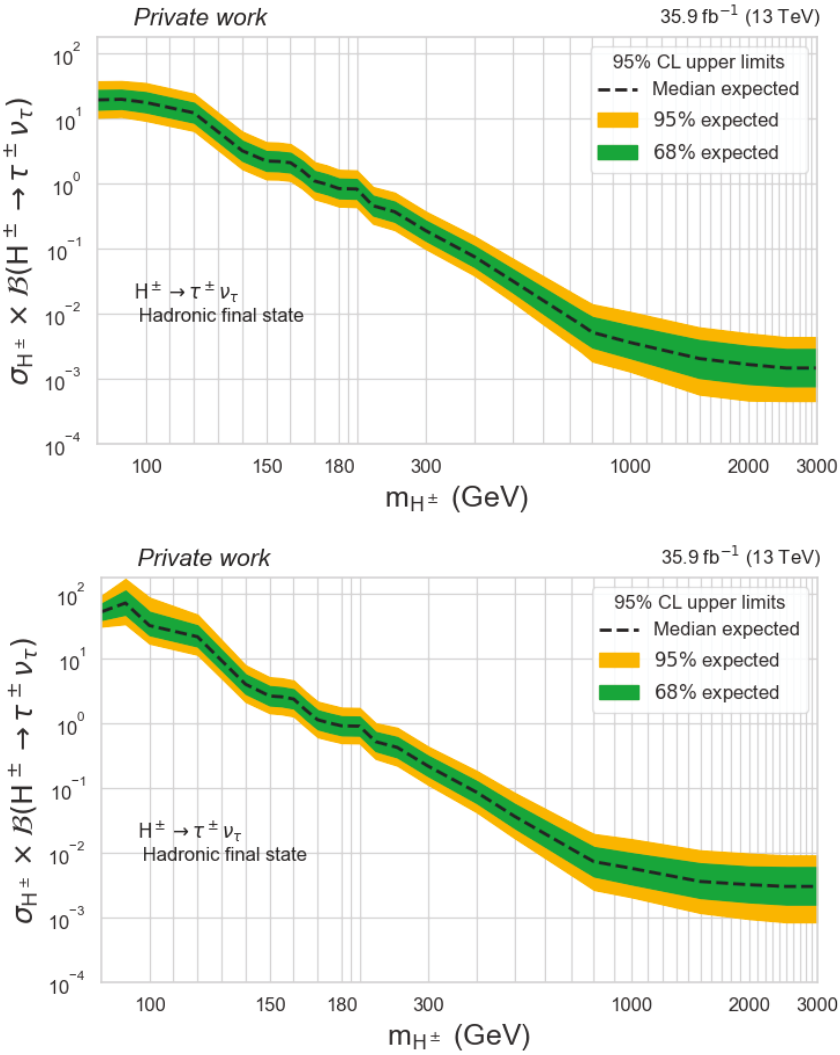


Figure 23.2: Expected 95% confidence level exclusion limits as a function of the charged Higgs boson mass $m_{H^{\pm}}$. **Top**: The baseline analysis. **Bottom**: The new analysis with the DNN selection step.

$m_{H^\pm}$	Expected limit					Observed
(GeV)	-2σ	-1σ	median	$+1\sigma$	$+2\sigma$	limit
80	31.62067	40.17137	53.08125	71.06590	94.00542	Blinded
90	34.51893	48.40075	72.73125	113.02872	171.32098	Blinded
100	17.28816	23.35382	32.66250	53.23317	88.15977	Blinded
120	11.59822	15.58694	21.91250	32.83052	48.01510	Blinded
140	2.17026	2.89356	4.04063	5.73177	7.90829	Blinded
150	1.45085	1.91469	2.66250	3.77685	5.17729	Blinded
155	1.39650	1.85466	2.58125	3.64101	5.00539	Blinded
160	1.29337	1.71770	2.39062	3.37213	4.60529	Blinded
165	0.88842	1.16800	1.61875	2.28335	3.09774	Blinded
170	0.61710	0.81955	1.14062	1.59983	2.17648	Blinded
175	0.55344	0.73037	1.01562	1.42451	1.95096	Blinded
180	0.50065	0.66070	0.91875	1.28863	1.75311	Blinded
200	0.49554	0.65396	0.90938	1.27548	1.73522	Blinded
220	0.28523	0.37642	0.52344	0.73834	1.00835	Blinded
250	0.22411	0.30082	0.42656	0.61530	0.85766	Blinded
300	0.11362	0.15458	0.21953	0.31666	0.43866	Blinded
400	0.04266	0.05921	0.08633	0.12865	0.18460	Blinded
500	0.01704	0.02438	0.03682	0.05721	0.08480	Blinded
750	0.00348	0.00549	0.00908	0.01509	0.02361	Blinded
800	0.00269	0.00436	0.00732	0.01229	0.01954	Blinded
1000	0.00207	0.00328	0.00576	0.00990	0.01627	Blinded
1500	0.00119	0.00202	0.00361	0.00690	0.01092	Blinded
2000	0.00096	0.00174	0.00322	0.00615	0.00974	Blinded
2500	0.00085	0.00160	0.00303	0.00602	0.00911	Blinded
3000	0.00085	0.00160	0.00303	0.00602	0.00911	Blinded

Table 23.2: Expected 95% CL exclusion limits for $\sigma_{H^\pm}\mathcal{B}(H^\pm \to \tau^\pm\nu_\tau)$ when the DNN selection is included in the analysis. The $\pm 1(2)\sigma$ corresponds to the 1 (2) standard deviation from the median value.

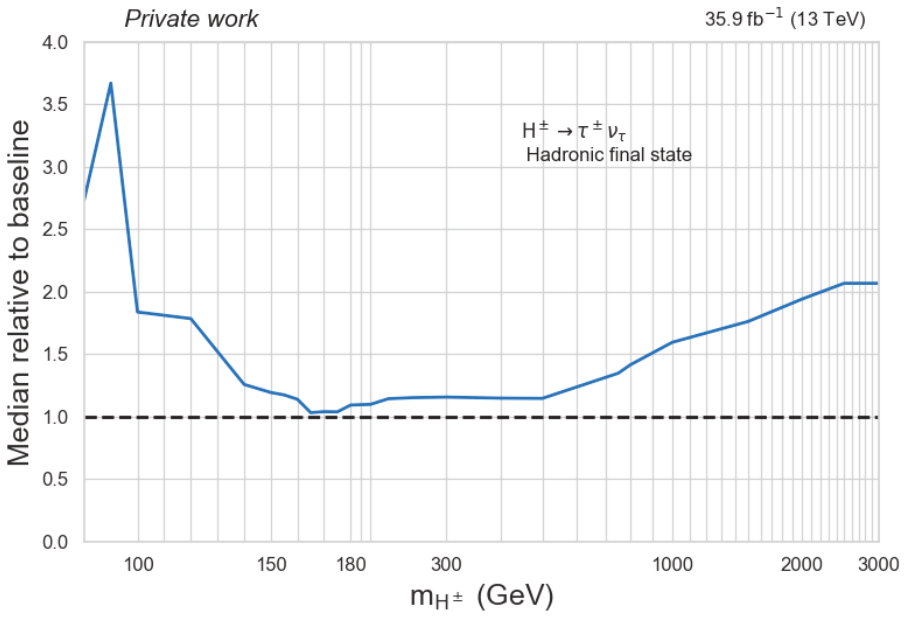


Figure 23.3: The expected limit median value after the new selection relative to the baseline analysis without the selection. We see a deterioration in the expected limits across the mass spectrum.

23.3 Discussion and future prospects

As is evident from the expected exclusion limits presented the addition of a trained deep neural network classifier to the event selection flow does not seem to provide a benefit to the charged Higgs boson search in this channel. However this approach of producing a transverse mass decorrelated classifier for pruning out the background events and continue using the transverse mass distribution as the summary statistic for computing the confidence limits might not be the optimal approach. In the following are some considerations of what might explain the lack of success with this approach and how to improve the subsequent attempts.

23.3.1 Choice of algorithm

While deep learning and various neural network algorithms are achieving fantastic results they have known downsides of being data intensive in their training requirements. Also they most often provide the astonishing results in realms where many other machine learning approaches are unusable due to a large dimensionality of the input variables. The environment where the deep neural network classifier was implemented in proved to be challenging in both regards. Especially due to the already optimized analysis flow the statistics that could be used for training the network were limited even when relaxing the actual cut values when gathering the training dataset.

A more natural placement for a deep neural network classifier could be as the only selection in the analysis where the input variables would represent lower level features. As is seen in the selection flow histogram 21.4 the requirement to have an identified b-jet represents a significant loss of signal. Removing the explicit requirement for a b-jet and replacing the input variables containing the reconstructed b-jet candidate variables with particle level information that is ultimately used to reconstruct the b-jets might empower the network to learn to pick up the signal events where a b-jet is not tagged with the current methods. Although it is to be noted that the modern jet taggers [233] already take advantage of the reconstructed particle level and event level information using a deep neural network to identify the b-jets so improving upon a jet classifier network specifically trained for the purpose might not be feasible in the context of this analysis.

Other methods that might work in this context could be the boosted decision tree algorithms. They tend to work well with high-level variables such as the ones used as input in this analysis but have the added benefit of requiring less data due to smaller number of trainable parameters controlling the algorithm. There is also some success in a similar analysis case in particle physics with this approach [234] however there the approach was to use the classifier output directly instead applying it just as a cut.

23.3.2 Columnar object based analysis framework

One difficulty faced in development and deployment of a deep neural network classifier arises from the event based loop approach taken in many particle physics analyses. This means the data is processed one event per thread and the analysis can be accelerated by increasing

the number of available processor cores. However the modern processor architectures have been developed to take advantage of vectorized operations where the more efficient approach would be to process batches of collision events at the same time, exploiting the data level parallelism of the problem. More concretely it is often faster to form a one dimensional vector of the p_{T}^{τ} values of hundreds of events and check which indices of the vector satisfy $p_{\mathrm{T}}^{\tau} \leq c$ for some constant c instead of creating a loop over the same collision events and checking for the condition individually.

The event based loop approach shows it weakness especially when running inference on a deep neural network for each event. Compared to a more common analysis step where the selection might be just evaluating a single condition, deep neural networks tend to apply a significantly larger number of operations per input in order to produce the output value. This can lead to the deep neural network classification step taking up more time than the rest of the event selection steps combined in the analysis, resulting in a slow turnaround time for evaluating network performance and hindering extensive testing of different networks.

One way to improve the throughput of this network inference step is to batch several entries together, allowing the process to spend less time on time consuming tasks like reloading the stored network parameter values for each individual entry separately. This can be done in the event based loops by implementing a waiting system where the events processed up until the deep neural network classification step, where the process waits until enough entries are aggregated to start a batched network inference but depending on the analysis framework such implementation can prove difficult to create in practice.

A more suitable approach enabling both the efficient use of vectorized operations as well as the batched inference of the deep neural networks would be to shift from the event based loop towards an array based approach. One such effort is the Columnar Object Framework For Effective Analysis (COFFEA) [235]. As the name suggests here the data is formatted into arrays where each row containing information from one event and each column containing some object or variable describing the event. This allows for a fast memory access to any single object of interest for multiple events as they can be allocated a contiguous memory block for storage. Such layout makes a batched inference for deep neural networks also more natural to implement and it is the default data format used in the popular deep learning frameworks. Moving onto array based analysis frameworks is likely to make deep learning based solutions easier and faster to implement into the analyses in the future.

23.4 Summary

The work presented in this chapter documents the first foray into using a deep neural network based classifier in the $H^{\pm} \to \tau^{\pm}\nu_{\tau}$ in the fully hadronic tau decay channel. While the result is a negative one and the additional deep neural network based selection step only worsens the expected 95% confidence level limits there is still value in experimentally demonstrating that the distance based correlation term does seem to drive the network training towards a decorrelated solution in a real analysis use case, uncovering some deep rooted efficiency issues with the event loop based analysis frameworks making them especially cumbersome to work with when evaluating machine learning based classifiers during the analysis and providing an

addition to the code base on how to implement a neural network that can be used in further studies.

While an improvement in the expected confidence limits would have been an indication for the approach to be working, the opposite is not a proof against the method. However it might encourage the exploration of other approaches.

Bibliography

[1] S. Glashow. "Partial-symmetries of weak interactions". In: *Nuclear Physics* 22.4 (Feb. 1961), pp. 579–588. DOI: 10.1016/0029-5582(61)90469-2.

[2] S. Weinberg. "A Model of Leptons". In: *Phys. Rev. Lett.* 19 (21 Nov. 1967), pp. 1264–1266. DOI: 10.1103/PhysRevLett.19.1264. URL: https://link.aps.org/doi/10.1103/PhysRevLett.19.1264.

[3] A. Salam and J.C. Ward. "Electromagnetic and weak interactions". In: *Physics Letters* 13.2 (1964), pp. 168–171. ISSN: 0031-9163. DOI: https://doi.org/10.1016/0031-9163(64)90711-5. URL: https://www.sciencedirect.com/science/article/pii/0031916364907115.

[4] M. Gell-Mann. "The Eightfold Way: A Theory of strong interaction symmetry". In: (Mar. 1961). DOI: 10.2172/4008239.

[5] G. Zweig. "An SU(3) model for strong interaction symmetry and its breaking. Version 2". In: *DEVELOPMENTS IN THE QUARK THEORY OF HADRONS. VOL. 1. 1964 - 1978*. Ed. by D. B. Lichtenberg and Simon Peter Rosen. Feb. 1964.

[6] H. Fritzsch, M. Gell-Mann, and H. Leutwyler. "Advantages of the color octet gluon picture". In: *Physics Letters B* 47.4 (1973), pp. 365–368. ISSN: 0370-2693. DOI: https://doi.org/10.1016/0370-2693(73)90625-4. URL: https://www.sciencedirect.com/science/article/pii/0370269373906254.

[7] G. Aad et al. "Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC". In: *Physics Letters B* 716.1 (Sept. 2012), pp. 1–29. ISSN: 0370-2693. DOI: 10.1016/j.physletb.2012.08.020. URL: http://dx.doi.org/10.1016/j.physletb.2012.08.020.

[8] The CMS Collaboration. "Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC". In: *Phys. Lett. Sect. B Nucl. Elem. Part. High-Energy Phys.* (2012). ISSN: 03702693. DOI: 10.1016/j.physletb.2012.08.021. arXiv: 1207.7235.

[9] A. Arbuzov. *Quantum Field Theory and the Electroweak Standard Model*. 2018. arXiv: 1801.05670 [hep-ph].

[10] F. Englert and R. Brout. "Broken symmetry and the mass of gauge vector mesons". In: *Phys. Rev. Lett.* 13.9 (1964), pp. 321–323. ISSN: 00319007. DOI: 10.1103/PhysRevLett.13.321.

[11] G. S. Guralnik, C. R. Hagen, and T. W.B. Kibble. "Global conservation laws and massless particles". In: *Phys. Rev. Lett.* 13.20 (1964), pp. 585–587. ISSN: 00319007. DOI: 10.1103/PhysRevLett.13.585.

[12] P. Higgs. "Broken symmetries and the masses of gauge bosons". In: *Phys. Rev. Lett.* 13.16 (1964), pp. 508–509. ISSN: 00319007. DOI: 10.1103/PhysRevLett.13.508.

[13] Nobel Media AB. *The Nobel Prize in Physics 2013*. URL: https://www.nobelprize.org/prizes/physics/2013/summary/.

[14] J. Goldstone. "Field theories with « Superconductor » solutions". In: *Nuovo Cim.* 1.19 (1961), pp. 154–164. ISSN: 00296341. DOI: 10.1007/BF02812722.

[15] P.Dirac. "The quantum theory of the emission and absorption of radiation". In: *Proc. R. Soc. London. Ser. A, Contain. Pap. a Math. Phys. Character* 114 (1927), pp. 243–265. ISSN: 0950-1207. DOI: 10.1098/rspa.1927.0039.

[16] S. Tomonaga. "On a Relativistically Invariant Formulation of the Quantum Theory of Wave Fields". In: *Prog. Theor. Phys.* 1.2 (1946), pp. 27–42. ISSN: 13474081. DOI: 10.1143/PTP.1.27.

[17] R. P. Feynman. "A relativistic cut-off for classical electrodynamics". In: *Phys. Rev.* 74.8 (1948), pp. 939–946. ISSN: 0031899X. DOI: 10.1103/PhysRev.74.939.

[18] J Schwinger. "Quantum electrodynamics. I. A covariant formulation". In: *Phys. Rev.* 74.10 (1948), pp. 1439–1461. ISSN: 0031899X. DOI: 10.1103/PhysRev.74.1439.

[19] T. Tati and S. Tomonaga. "A Self-Consistent Subtraction Method in the Quantum Field Theory, I". In: *Prog. Theor. Phys.* 3.4 (1948), pp. 391–406. ISSN: 0033-068X. DOI: 10.1143/ptp/3.4.391.

[20] P.Dirac. "The quantum theory of the electron". In: *Proc. R. Soc. London. Ser. A, Contain. Pap. a Math. Phys. Character* 117 (1928), pp. 610–624. ISSN: 0950-1207. DOI: 10.1098/rspa.1928.0023.

[21] Gregory Ciezarek et al. *A challenge to lepton universality in B-meson decays*. 2017. DOI: 10.1038/nature22346. arXiv: 1703.01766.

[22] F. Wilson. "Fermi's Theory of Beta Decay (translation of the original)". In: *Am. J. Phys.* 36.12 (1968), pp. 1150–1160. ISSN: 0002-9505. DOI: 10.1119/1.1974382.

[23] T. D. Lee and C. N. Yang. "Question of parity conservation in weak interactions". In: *Phys. Rev.* 104.1 (1956), pp. 254–258. ISSN: 0031899X. DOI: 10.1103/PhysRev.104.254.

[24] J. H. Christenson et al. "Evidence for the 2π decay of the K20 meson". In: *Phys. Rev. Lett.* 13.4 (1964), pp. 138–140. ISSN: 00319007. DOI: 10.1103/PhysRevLett.13.138.

[25] J. Bardeen, L. N. Cooper, and J. R. Schrieffer. *Microscopic theory of superconductivity*. 1957. DOI: 10.1103/PhysRev.106.162.

[26] Y. Nambu and G. Jona-Lasinio. "Dynamical model of elementary particles based on an analogy with superconductivity. i". In: *Phys. Rev.* 122.1 (1961), pp. 345–358. ISSN: 0031899X. DOI: 10.1103/PhysRev.122.345.

[27] Y. Nambu and G. Jona-Lasinio. "Dynamical model of elementary particles based on an analogy with superconductivity. II". In: *Phys. Rev.* 124.1 (1961), pp. 246–254. ISSN: 0031899X. DOI: 10.1103/PhysRev.124.246.

[28] E. D. Bloom et al. "High-energy inelastic e-p scattering at 6°and 10°". In: *Phys. Rev. Lett.* 23.1 (1969), pp. 930–934. ISSN: 00319007. DOI: 10.1103/PhysRevLett.23.930.

[29] M. Breidenbach et al. "Observed behavior of highly inelastic electron-proton scattering". In: *Phys. Rev. Lett.* 23.16 (1969), pp. 935–939. ISSN: 00319007. DOI: 10.1103/PhysRevLett.23.935.

[30] D. Gross and F. Wilczek. "Ultraviolet behavior of non-abelian gauge theories". In: *Phys. Rev. Lett.* 30.26 (1973), pp. 1343–1346. ISSN: 00319007. DOI: 10.1103/PhysRevLett.30.1343.

[31] H. David Politzer. "Reliable perturbative results for strong interactions?" In: *Phys. Rev. Lett.* 30.26 (1973), pp. 1346–1349. ISSN: 00319007. DOI: 10.1103/PhysRevLett.30.1346.

[32] V. N. Gribov and L. N. Lipatov. "Deep inelastic e p scattering in perturbation theory". In: *Sov. J. Nucl. Phys.* 15 (1972), pp. 438–450.

[33] Y. Dokshitzer. "Calculation of the Structure Functions for Deep Inelastic Scattering and e+ e- Annihilation by Perturbation Theory in Quantum Chromodynamics." In: *Sov. Phys. JETP* 46 (1977), pp. 641–653.

[34] G. Altarelli and G. Parisi. "Asymptotic freedom in parton language". In: *Nucl. Physics, Sect. B* 126 (1977), pp. 298–318. ISSN: 05503213. DOI: 10.1016/0550-3213(77)90384-4.

[35] NNPDF Collaboration. "Parton distributions for the LHC run II". In: *J. High Energy Phys.* (2015). ISSN: 10298479. DOI: 10.1007/JHEP04(2015)040. arXiv: 1410.8849.

[36] T. Sjostrand et al. "An introduction to PYTHIA 8.2". In: *Comput. Phys. Commun.* (2015). ISSN: 00104655. DOI: 10.1016/j.cpc.2015.01.024. arXiv: 1410.3012.

[37] S. Sjostrand, S. Torbjorn, and P. Mrenna. "PYTHIA 6.4 physics and manual". In: (2006). ISSN: 11266708.

[38] G. et al. Corcella. "HERWIG 6.5 Release Note". In: *arXiv Prepr. hep-ph/0210213* (2002).

[39] S. Hoche. "Introduction to Parton-Shower Event Generators". In: 2015, pp. 235–295. DOI: 10.1142/9789814678766_0005. arXiv: 1411.4085.

[40] A. et al Buckley. *General-purpose event generators for LHC physics*. 2011. DOI: 10.1016/j.physrep.2011.03.005. arXiv: 1101.2599.

[41] D. Sakharov. "Violation of cp in variance, C asymmetry, and baryon asymmetry of the universe". In: *Sov. Phys. - Uspekhi* 34.5 (1991), pp. 392–393. ISSN: 21695296. DOI: 10.1070/PU1991v034n05ABEH002497.

[42] Particle Data Group. "Review of particle physics". In: *Phys. Rev. D* (2018). ISSN: 2470-0010.

[43] S. Martin. "A Supersymmetry Primer". In: 1998, pp. 30–51. DOI: 10.1142/9789812839657_0001. arXiv: 9709356 [hep-ph].

[44] P. Fayet. "Spontaneously broken supersymmetric theories of weak, electromagnetic and strong interactions". In: *Phys. Lett. B* 69.4 (1977), pp. 489–494. ISSN: 03702693. DOI: 10.1016/0370-2693(77)90852-8.

[45] T. D. Lee. "A theory of spontaneous t violation". In: *Phys. Rev. D* 8 (1973), pp. 1226–1239. ISSN: 05562821. DOI: 10.1103/PhysRevD.8.1226.

[46] N. Blinov, S. Profumo, and T. Stefaniak. "The electroweak phase transition in the Inert Doublet Model". In: *J. Cosmol. Astropart. Phys.* 7 (2015), pp. 28–28. ISSN: 14757516. DOI: 10.1088/1475-7516/2015/07/028. arXiv: 1504.05949.

[47] G. C. Dorsch, S. J. Huber, and J. M. No. "A strong electroweak phase transition in the 2HDM after LHC8". In: *J. High Energy Phys.* 2013 (2013), p. 29. ISSN: 10298479. DOI: 10.1007/JHEP10(2013)029. arXiv: 1305.6610.

[48] J. et al. Andersen. “Nonperturbative Analysis of the Electroweak Phase Transition in the Two Higgs Doublet Model”. In: *Phys. Rev. Lett.* 121.19 (2018), p. 6. ISSN: 10797114. DOI: 10.1103/PhysRevLett.121.191802. arXiv: 1711.09849.

[49] M. Cauchy. “Méthode générale pour la résolution des systèmes d’équations simultanées. Übersetzt von Richard Pulskamp, 2010.” In: *Compte rendu des séances l’académie des Sci.* (1847).

[50] F. Rosenblatt. “The perceptron: A probabilistic model for information storage and organization in the brain”. In: *Psychol. Rev.* 65.6 (1958), pp. 386–408. ISSN: 0033295X. DOI: 10.1037/h0042519.

[51] H. Robbins and S. Monro. “A Stochastic Approximation Method”. In: *Ann. Math. Stat.* 22.3 (1951), pp. 400–407. ISSN: 0003-4851. DOI: 10.1214/aoms/1177729586.

[52] L. Bottou and O. Bousquet. “The tradeoffs of large scale learning”. In: *Adv. Neural Inf. Process. Syst. 20 - Proc. 2007 Conf.* 2009. ISBN: 160560352X.

[53] D. Rumelhart, G. Hinton, and R. Williams. “Learning representations by back-propagating errors”. In: *Nature* 323 (1986), pp. 533–536. ISSN: 00280836. DOI: 10.1038/323533a0.

[54] Honda. “ The Honda Prize 2019 Awarded to Dr. Geoffrey Hinton, Professor Emeritus, the University of Toronto and Chief Scientific Adviser, Vector Institute”. In: (20.09.2019). URL: https://global.honda/newsroom/news/2019/c190920eng.html.

[55] J. Schmidhuber. “Critique of Honda Prize for Dr. Hinton”. In: (21.04.2020). URL: http://people.idsia.ch/~juergen/critique-honda-prize-hinton.html.

[56] S. Linnainmaa. “The representation of the cumulative rounding error of an algorithm as a Taylor expansion of the local rounding errors”. In: *Master’s thesis, University of Helsinki* (1970).

[57] Papers with Code. “Image classification on ImageNet - Leaderboard”. In: (17.05.2020). URL: https://paperswithcode.com/sota/image-classification-on-imagenet.

[58] T. Karras et al. “Progressive growing of GANs for improved quality, stability, and variation”. In: *6th Int. Conf. Learn. Represent. ICLR 2018 - Conf. Track Proc.* 2018. arXiv: 1710.10196.

[59] E. Yurtsever et al. “A Survey of Autonomous Driving: Common Practices and Emerging Technologies”. In: *IEEE Access* (2020). ISSN: 21693536. DOI: 10.1109/ACCESS.2020.2983149. arXiv: 1906.05113.

[60] V. Nair and G. E. Hinton. “Rectified linear units improve Restricted Boltzmann machines”. In: *ICML 2010 - Proceedings, 27th Int. Conf. Mach. Learn.* 2010. ISBN: 9781605589077.

[61] Y. LeCun et al. “Backpropagation Applied to Handwritten Zip Code Recognition”. In: *Neural Comput.* 1.4 (1989), pp. 541–551. ISSN: 0899-7667. DOI: 10.1162/neco.1989.1.4.541.

[62] K. Hornik, M. Stinchcombe, and H. White. “Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks”. In: *Neural Networks* 3.5 (1990), pp. 551–560. ISSN: 08936080. DOI: 10.1016/0893-6080(90)90005-6.

[63] Z. Lu et al. “The expressive power of neural networks: A view from the width”. In: *Adv. Neural Inf. Process. Syst.* 2017. arXiv: 1709.02540.

[64] et al. A. Paszke. “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach et al. Curran Associates, Inc., 2019, pp. 8024–8035. URL: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[65] A. Martıén et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015. URL: https://www.tensorflow.org/.

[66] S. Hochreiter. “Untersuchungen zu dynamischen neuronalen Netzen”. In: *Master's thesis, Inst. für Inform. Tech. Univ. Munchen* (1991). ISSN: 18168957 18163459. arXiv: 1511.07289.

[67] C. Zhang et al. “Deep Learning Requires Rethinking Generalization”. In: *Int. Conf. Learn. Represent.* (2017). arXiv: 1611.03530.

[68] R. Tibshirani. “Regression Shrinkage and Selection Via the Lasso”. In: *J. R. Stat. Soc. Ser. B* 58.1 (1996), pp. 267–288. ISSN: 0035-9246. DOI: 10.1111/j.2517-6161.1996.tb02080.x.

[69] A N Tikhonov. “On the stability of inverse problems”. In: *Dokl. Akad. Nauk Sssr* 39 (1943), pp. 195–198.

[70] A. N. Tikhonov. “Solution of incorrectly formulated problems and the regularization method”. In: *Dokl. Akad. Nauk SSSR* (1963).

[71] J. Bell, A. N. Tikhonov, and V. Y. Arsenin. “Solutions of Ill-Posed Problems.” In: *Math. Comput.* 21.2 (1978), pp. 266–267. ISSN: 00255718. DOI: 10.2307/2006360.

[72] A. N. Tikhonov et al. *Numerical Methods for the Solution of Ill-Posed Problems*. 1995. DOI: 10.1007/978-94-015-8480-7.

[73] N. Srivastava et al. “Dropout: A simple way to prevent neural networks from overfitting”. In: *J. Mach. Learn. Res.* 15.1 (2014), pp. 1929–1958. ISSN: 15337928.

[74] World Economic Forum. “How to Prevent Discriminatory Outcomes in Machine Learning”. In: *Glob. Futur. Counc. Hum. Rights 2016-2018* March 2018.March (2016), p. 29. ISSN: 1936878X. DOI: 10.1016/j.jcmg.2009.08.003. URL: http://linkinghub.elsevier.com/retrieve/pii/S1936878X09005683.

[75] S. Wunsch et al. “Reducing the dependence of the neural network function to systematic uncertainties in the input space”. In: (2019). DOI: 10.1007/s41781-020-00037-9. arXiv: 1907.11674. URL: http://arxiv.org/abs/1907.11674.

[76] P. Englert C.and Galler, P. Harris, and M. Spannowsky. “Machine learning uncertainties with adversarial neural networks”. In: *Eur. Phys. J. C* 79.1 (2019), pp. 1–10. ISSN: 14346052. DOI: 10.1140/epjc/s10052-018-6511-8. arXiv: 1807.08763.

[77] C. Shimmin et al. “Decorrelated jet substructure tagging using adversarial neural networks”. In: *Phys. Rev. D* 96.7 (2017). ISSN: 24700029. DOI: 10.1103/PhysRevD.96.074034. arXiv: 1703.03507.

[78] G. Louppe, M. Kagan, and K. Cranmer. “Learning to pivot with adversarial networks”. In: *Adv. Neural Inf. Process. Syst.* 2017-Decem.Nips (2017), pp. 982–991. ISSN: 10495258. arXiv: 1611.01046.

[79] S. Kullback and R. A. Leibler. “On Information and Sufficiency”. In: *Ann. Math. Stat.* 22.1 (1951), pp. 79–86. ISSN: 0003-4851. DOI: 10.1214/aoms/1177729694.

[80] J. Lin. "Divergence Measures Based on the Shannon Entropy". In: *IEEE Trans. Inf. Theory* 37.1 (1991), pp. 145–151. ISSN: 15579654. DOI: 10.1109/18.61115.

[81] S. Wunsch et al. "Reducing the Dependence of the Neural Network Function to Systematic Uncertainties in the Input Space". In: *Comput. Softw. Big Sci.* 4.1 (2020), p. 5. ISSN: 2510-2044. DOI: 10.1007/s41781-020-00037-9. URL: https://doi.org/10.1007/s41781-020-00037-9.

[82] G. Kasieczka and D. Shih. "DisCo Fever: Robust Networks Through Distance Correlation". In: (2020), pp. 1–9. arXiv: 2001.05310. URL: http://arxiv.org/abs/2001.05310.

[83] S. Chang, T. Cohen, and B. Ostdiek. "What is the machine learning?" In: *Phys. Rev. D* 97.5 (2018). ISSN: 24700029. DOI: 10.1103/PhysRevD.97.054024. arXiv: 1709.10106.

[84] L. Bradshaw et al. "Mass agnostic jet taggers". In: *SciPost Phys.* 8.1 (2020), p. 11. ISSN: 2542-4653. DOI: 10.21468/scipostphys.8.1.011.

[85] Y. Ganin and V. Lempitsky. "Unsupervised domain adaptation by backpropagation". In: *32nd Int. Conf. Mach. Learn. ICML 2015*. 2015, pp. 1180–1189. ISBN: 9781510810587. arXiv: 1409.7495.

[86] I. Goodfellow et al. "Generative adversarial nets". In: *Adv. Neural Inf. Process. Syst.* 2014, pp. 2672–2680.

[87] M. Arjovsky, S. Chintala, and L. Bottou. "Wasserstein generative adversarial networks". In: *34th Int. Conf. Mach. Learn. ICML 2017*. 2017, pp. 214–223. ISBN: 9781510855144.

[88] I. Gulrajani et al. "Improved training of wasserstein GANs". In: *Adv. Neural Inf. Process. Syst.* 2017, pp. 5769–5779. arXiv: 1704.00028.

[89] T. Salimans et al. "Improved techniques for training GANs". In: *Adv. Neural Inf. Process. Syst.* 2016, pp. 2234–2242. arXiv: 1606.03498.

[90] M. Szekely G.and Rizzo and N. Bakirov. "Measuring and testing dependence by correlation of distances". In: *Ann. Stat.* 35.6 (2007), pp. 2769–2794. ISSN: 00905364. DOI: 10.1214/009053607000000505.

[91] G. Szekely and M. Rizzo. "Brownian distance covariance". In: *Ann. Appl. Stat.* 3.4 (2009), pp. 1236–1265. ISSN: 19326157. DOI: 10.1214/09-AOAS312.

[92] G. J. Szekely and M. Rizzo. "The distance correlation t -test of independence in high dimension". In: *J. Multivar. Anal.* 117 (2013), pp. 193–213. ISSN: 0047259X. DOI: 10.1016/j.jmva.2013.02.012. URL: http://dx.doi.org/10.1016/j.jmva.2013.02.012.

[93] Gábor J. Székely and Maria L. Rizzo. "Partial distance correlation with methods for dissimilarities". In: *Ann. Stat.* 42.6 (2014), pp. 2382–2412. ISSN: 00905364. DOI: 10.1214/14-AOS1255. arXiv: 1310.2926.

[94] G. Szekely and M. Rizzo. "On the uniqueness of distance covariance". In: *Stat. Probab. Lett.* 82.12 (2012), pp. 2278–2282. ISSN: 01677152. DOI: 10.1016/j.spl.2012.08.007.

[95] A. Andreassen and B. Nachman. "Neural networks for full phase-space reweighting and parameter tuning". In: *Phys. Rev. D* (2020). ISSN: 24700029. DOI: 10.1103/PhysRevD.101.091901. arXiv: 1907.08209.

[96] O. Bruning and L. Rossi. *The High Luminosity Large Hadron Collider*. WORLD SCIENTIFIC, 2015. DOI: 10.1142/9581. eprint: https://www.worldscientific.com/

doi/pdf/10.1142/9581. URL: https://www.worldscientific.com/doi/abs/10.1142/9581.

[97] K. Albertsson et al. "Machine Learning in High Energy Physics Community White Paper". In: *J. Phys. Conf. Ser.* 2018. DOI: 10.1088/1742-6596/1085/2/022008. arXiv: 1807.02876.

[98] S. Forte et al. "Neural network parametrization of deep-inelastic structure functions". In: *J. High Energy Phys.* 2002.5 (2002), pp. 62–62. ISSN: 10298479. DOI: 10.1088/1126-6708/2002/05/062. arXiv: 0204232 [hep-ph].

[99] R. Ball et al. "A first unbiased global NLO determination of parton distributions and their uncertainties". In: *Nucl. Phys. B* 838.1-2 (2010), 136–206. ISSN: 05503213. DOI: 10.1016/j.nuclphysb.2010.05.008.

[100] J. Duarte et al. "Fast inference of deep neural networks in FPGAs for particle physics". In: *J. Instrum.* 13.7 (2018). ISSN: 17480221. DOI: 10.1088/1748-0221/13/07/P07027. arXiv: 1804.06913.

[101] "Boosted Decision Trees in the Level-1 Muon Endcap Trigger at CMS". In: *J. Phys. Conf. Ser.* 2018. DOI: 10.1088/1742-6596/1085/4/042042.

[102] O. Cerri et al. "Variational autoencoders for new physics mining at the Large Hadron Collider". In: *J. High Energy Phys.* 2019.5 (2019). ISSN: 10298479. DOI: 10.1007/JHEP05(2019)036. arXiv: 1811.10276.

[103] M. Borisyak et al. "Towards automation of data quality system for CERN CMS experiment". In: *J. Phys. Conf. Ser.* 2017. DOI: 10.1088/1742-6596/898/9/092041. arXiv: 1709.08607.

[104] A. Pol et al. "Detector Monitoring with Artificial Neural Networks at the CMS Experiment at the CERN Large Hadron Collider". In: *Comput. Softw. Big Sci.* (2019). ISSN: 2510-2036. DOI: 10.1007/s41781-018-0020-1. arXiv: 1808.00911.

[105] R. Rahmat, R. Kroeger, and A. Giammanco. "The fast simulation of the CMS experiment". In: *J. Phys. Conf. Ser.* 2012. DOI: 10.1088/1742-6596/396/6/062016.

[106] T. Karras, S. Laine, and T. Aila. "A style-based generator architecture for generative adversarial networks". In: *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.* 2019. ISBN: 9781728132938. DOI: 10.1109/CVPR.2019.00453. arXiv: 1812.04948.

[107] L. de Oliveira, M. Paganini, and B. Nachman. "Learning Particle Physics by Example: Location-Aware Generative Adversarial Networks for Physics Synthesis". In: *Comput. Softw. Big Sci.* (2017). ISSN: 2510-2036. DOI: 10.1007/s41781-017-0004-6. arXiv: 1701.05927.

[108] M. Paganini, L. De Oliveira, and B. Nachman. "Accelerating Science with Generative Adversarial Networks: An Application to 3D Particle Showers in Multilayer Calorimeters". In: *Phys. Rev. Lett.* (2018). ISSN: 10797114. DOI: 10.1103/PhysRevLett.120.042003. arXiv: 1705.02355.

[109] R. Sipio et al. "A generative-adversarial network approach for the simulation of QCD dijet events at the LHC". In: *Proc. Sci.* 2019. DOI: 10.22323/1.367.0050.

[110] Sofia Vallecorsa, Federico Carminati, and Gulrukh Khattak. "3D convolutional GAN for fast simulation". In: *EPJ Web Conf.* 214.1 (2019). DOI: 10.1051/epjconf/201921402010.

[111] J. Albrecht et al. “A Roadmap for HEP Software and Computing R&D for the 2020s”. In: *Comput. Softw. Big Sci.* 3.1 (2019). ISSN: 2510-2036. DOI: 10.1007/s41781-018-0018-8. arXiv: 1712.06982.

[112] S. Farrell et al. “The HEP.TrkX Project: Deep neural networks for HL-LHC online and offline tracking”. In: *EPJ Web Conf.* 2017. DOI: 10.1051/epjconf/201715000003.

[113] Giuseppe Cerati et al. “Kalman filter tracking on parallel architectures”. In: *J. Phys. Conf. Ser.* 2015. DOI: 10.1088/1742-6596/664/7/072008.

[114] G. Cerati et al. “Kalman-Filter-based particle tracking on parallel architectures at Hadron Colliders”. In: *2015 IEEE Nucl. Sci. Symp. Med. Imaging Conf. NSS/MIC 2015*. 2016. ISBN: 9781467398626. DOI: 10.1109/NSSMIC.2015.7581932. arXiv: 1601.08245.

[115] G. Cerati et al. “Kalman filter tracking on parallel architectures”. In: *J. Phys. Conf. Ser.* 2017. DOI: 10.1088/1742-6596/898/4/042051.

[116] G. Cerati et al. “Parallelized Kalman-Filter-Based Reconstruction of Particle Tracks on Many-Core Architectures”. In: *J. Phys. Conf. Ser.* 2018. DOI: 10.1088/1742-6596/1085/4/042016. arXiv: 1711.06571.

[117] G. Cerati et al. “Parallelized and Vectorized Tracking Using Kalman Filters with CMS Detector Geometry and Events”. In: *EPJ Web Conf.* (2019). DOI: 10.1051/epjconf/201921402002. arXiv: 1811.04141.

[118] G. Cerati et al. *Reconstruction of Charged Particle Tracks in Realistic Detector Geometry Using a Vectorized and Parallelized Kalman Filter Algorithm*. 2020. arXiv: 2002.06295 [physics.ins-det].

[119] A. Tsaris et al. “The HEP.TrkX Project: Deep Learning for Particle Tracking”. In: *J. Phys. Conf. Ser.* 2018. DOI: 10.1088/1742-6596/1085/4/042023.

[120] X. Ju et al. *Graph Neural Networks for Particle Reconstruction in High Energy Physics detectors*. 2020. arXiv: 2003.11603 [physics.ins-det].

[121] S. Farrell et al. *Novel deep learning methods for track reconstruction*. 2018. arXiv: 1810.06111 [hep-ex].

[122] A. Florio, F. Pantaleo, and A. Carta. “Convolutional Neural Network for Track Seed Filtering at the CMS High-Level Trigger”. In: *J. Phys. Conf. Ser.* 2018. DOI: 10.1088/1742-6596/1085/4/042040.

[123] M. et al. De Cian. *Fast neural-net based fake track rejection in the LHCb reconstruction*. Tech. rep. Geneva: CERN, Mar. 2017. URL: https://cds.cern.ch/record/2255039.

[124] G. Aad. “A neural network clustering algorithm for the ATLAS silicon pixel detector”. In: *Journal of Instrumentation* 9.09 (Sept. 2014). DOI: 10.1088/1748-0221/9/09/P09009.

[125] The CMS Collaboration. “Performance of photon reconstruction and identification with the CMS detector in proton-proton collisions at s = 8 TeV”. In: *J. Instrum.* 10 (2015). ISSN: 17480221. DOI: 10.1088/1748-0221/10/08/P08010. arXiv: 1502.02702.

[126] The CMS Collaboration. “Reconstruction and identification of τ lepton decays to hadrons and $\nu\tau$ at CMS”. In: *J. Instrum.* 11 (2016). ISSN: 17480221. DOI: 10.1088/1748-0221/11/01/P01019.

[127] The CMS Collaboration. "Observation of Higgs Boson Decay to Bottom Quarks". In: *Phys. Rev. Lett.* 121.12 (2018). ISSN: 10797114. DOI: 10.1103/PhysRevLett.121.121801. arXiv: 1808.08242.

[128] M. Verzetti. "Machine learning techniques for jet flavour identification at CMS". In: *EPJ Web Conf.* (2019). DOI: 10.1051/epjconf/201921406010.

[129] H. Qu and L. Gouskos. "Jet tagging via particle clouds". In: *Phys. Rev. D* 101.5 (2020). ISSN: 24700029. DOI: 10.1103/PhysRevD.101.056019.

[130] J. Cogan et al. "Jet-images: computer vision inspired techniques for jet tagging". In: *J. High Energy Phys.* 2015.2 (2015). ISSN: 10298479. DOI: 10.1007/JHEP02(2015)118.

[131] CMS Collaboration. *A deep neural network for simultaneous estimation of b jet energy and resolution*. 2019. arXiv: 1912.06046 [hep-ex].

[132] Delphi Collaboration. "Classification of the hadronic decays of the Z0 into b and c quark pairs using a neural network". In: *Phys. Lett. B* 295.3 (1992), pp. 383–395. ISSN: 03702693. DOI: 10.1016/0370-2693(92)91580-3.

[133] The CMS Collaboration. "Observation of the diphoton decay of the Higgs boson and measurement of its properties". In: *Eur. Phys. J. C* 74.10 (2014). ISSN: 14346052. DOI: 10.1140/epjc/s10052-014-3076-z. arXiv: 1407.0558.

[134] L. Evans and P. Bryant. "LHC Machine". In: *J. Instrum.* (2008). ISSN: 17480221. DOI: 10.1088/1748-0221/3/08/S08001.

[135] E. Mobs. "The CERN accelerator complex - 2019. Complexe des accélérateurs du CERN - 2019". In: (July 2019). General Photo. URL: https://cds.cern.ch/record/2684277.

[136] W.J. Sterling. *Standard Model cross sections as a function of collider energy, with 125 GeV Higgs*. Accessed 1 June 2020. URL: http://www.hep.ph.ic.ac.uk/~wstirlin/plots/plots.html.

[137] The CMS Collaboration. *Cumulative delivered and recorded luminosity versus time for 2010-2012 and 2015-2018 (pp data only)*. Accessed 1 June 2020. URL: https://twiki.cern.ch/twiki/bin/view/CMSPublic/LumiPublicResults#Multi_year_plots.

[138] M. Giovannozzi and F. F. Van der Veken. "Description of the luminosity evolution for the CERN LHC including dynamic aperture effects, Part I: The model". In: *Nucl. Instruments Methods Phys. Res. Sect. A Accel. Spectrometers, Detect. Assoc. Equip.* (2018). ISSN: 01689002. DOI: 10.1016/j.nima.2018.07.063.

[139] M. Giovannozzi and F. F. Van der Veken. "Description of the luminosity evolution for the CERN LHC including dynamic aperture effects. Part II: application to Run 1 data". In: *Nucl. Instruments Methods Phys. Res. Sect. A Accel. Spectrometers, Detect. Assoc. Equip.* (2018). ISSN: 01689002. DOI: 10.1016/j.nima.2018.08.019.

[140] J. Blewett. "200-GeV intersecting storage accelerators". In: (1971). URL: https://cds.cern.ch/record/1068131.

[141] "CERN Press Release: CERN experiments observe particle consistent with long-sought Higgs boson. Communiqué de presse du CERN : Les expériences du CERN observent une particule dont les caractéristiques sont compatibles avec celles du boson de Higgs tant attendu". In: BUL-NA-2012-222. 28/2012 (July 2012), p. 1. URL: https://cds.cern.ch/record/1459454.

[142] ATLAS Collaboration. "Observation of H→bb decays and VH production with the ATLAS detector". In: *Phys. Lett. Sect. B Nucl. Elem. Part. High-Energy Phys.* (2018). ISSN: 03702693. DOI: 10.1016/j.physletb.2018.09.013.

[143] LHCb Collaboration. "Observation of J /ψp Resonances Consistent with Pentaquark States in Λb0 → j /ψK-p Decays". In: *Phys. Rev. Lett.* (2015). ISSN: 10797114. DOI: 10.1103/PhysRevLett.115.072001.

[144] The CMS Collaboration. "Measurement of the top quark mass in the all-jets final state at $\sqrt{s}$=13TeV and combination with the lepton+jets channel". In: *Eur. Phys. J. C* (2019). ISSN: 14346052. DOI: 10.1140/epjc/s10052-019-6788-2.

[145] The CMS Collaboration. "Observation and studies of jet quenching in PbPb collisions at sNN=2.76 TeV". In: *Phys. Rev. C - Nucl. Phys.* (2011). ISSN: 1089490X. DOI: 10.1103/PhysRevC.84.024906.

[146] G. Apollinari et al. *High Luminosity Large Hadron Collider HL-LHC*. 2017. arXiv: 1705.08830 [physics.acc-ph].

[147] ATLAS and CMS Collaborations. *Report on the Physics at the HL-LHC and Perspectives for the HE-LHC*. 2019. arXiv: 1902.10229 [hep-ex].

[148] CMS Collaboration. *CMS detector design illustration*. Accessed 3 June 2020. URL: http://cms.web.cern.ch/news/cms-detector-design.

[149] S. Davis. "Interactive Slice of the CMS detector". In: (Aug. 2016). URL: http://cds.cern.ch/record/2205172.

[150] CMS Collaboration. *CMS Phase1 tracking system in r-z view*. Accessed 5 June 2020. URL: https://twiki.cern.ch/twiki/bin/view/CMSPublic/DPGResultsTRK.

[151] The CMS Collaboration. "Observation of the diphoton decay of the Higgs boson and measurement of its properties". In: *Eur. Phys. J. C* 74.10 (2014). ISSN: 14346052. DOI: 10.1140/epjc/s10052-014-3076-z. arXiv: 1407.0558.

[152] The CMS Collaboration. "Performance and operation of the CMS electromagnetic calorimeter". In: *J. Instrum.* (2010). ISSN: 17480221. DOI: 10.1088/1748-0221/5/03/T03010.

[153] C. Biino. "The CMS Electromagnetic Calorimeter: Overview, lessons learned during Run 1 and future projections". In: *J. Phys. Conf. Ser.* 2015. DOI: 10.1088/1742-6596/587/1/012001.

[154] K. Zhu. "The CMS ECAL laser monitoring system". In: *IEEE Nucl. Sci. Symp. Conf. Rec.* 2007. ISBN: 1424409233. DOI: 10.1109/NSSMIC.2007.4436306.

[155] A. Martelli. "Evolution of the response of the CMS ECAL and possible design options for electromagnetic calorimetry at the HL-LHC". In: *J. Instrum.* (2014). ISSN: 17480221. DOI: 10.1088/1748-0221/9/04/C04017.

[156] The CMS Collaboration. "The CMS experiment at the CERN LHC". In: *J. Instrum.* (2008). ISSN: 17480221. DOI: 10.1088/1748-0221/3/08/S08004.

[157] G. Negro. "Performance of the CMS precision electromagnetic calorimeter at the LHC Run II and prospects for high-luminosity LHC". In: *Nucl. Instruments Methods Phys. Res. Sect. A Accel. Spectrometers, Detect. Assoc. Equip.* (2018). ISSN: 01689002. DOI: 10.1016/j.nima.2017.10.079.

[158] The CMS Collaboration. "Performance of the CMS hadron calorimeter with cosmic ray muons and LHC beam data". In: *J. Instrum.* (2010). ISSN: 17480221. DOI: 10.1088/1748-0221/5/03/T03012.

[159] The CMS Collaboration. "Performance of the CMS muon detector and muon reconstruction with proton-proton collisions at s=13 TeV". In: *J. Instrum.* (2018). ISSN: 17480221. DOI: 10.1088/1748-0221/13/06/P06015. arXiv: 1804.04528.

[160] The CMS Collaboration. "The CMS trigger system". In: *J. Instrum.* (2017). ISSN: 17480221. DOI: 10.1088/1748-0221/12/01/P01020. arXiv: 1609.02366.

[161] M. Tosi. *The CMS trigger in Run 2*. Tech. rep. CMS-CR-2017-340. Geneva: CERN, Oct. 2017. DOI: 10.22323/1.314.0523. URL: https://cds.cern.ch/record/2290106.

[162] The CMS Collaboration. "Particle-flow reconstruction and global event description with the CMS detector". In: *J. Instrum.* (2017). ISSN: 17480221. DOI: 10.1088/1748-0221/12/10/P10003. arXiv: 1706.04965.

[163] K. Rose. "Deterministic annealing for clustering, compression, classification, regression, and related optimization problems". In: *Proc. IEEE* (1998). ISSN: 00189219. DOI: 10.1109/5.726788.

[164] Wolfgang Waltenberger, Rudolf Frühwirth, and Pascal Vanlaer. "Adaptive vertex fitting". In: *J. Phys. G Nucl. Part. Phys.* (2007). ISSN: 09543899. DOI: 10.1088/0954-3899/34/12/N01.

[165] The CMS Collaboration. "Description and performance of track and primary-vertex reconstruction with the CMS tracker". In: *J. Instrum.* (2014). ISSN: 17480221. DOI: 10.1088/1748-0221/9/10/P10009. arXiv: 1405.6569.

[166] "GEANT4 - A simulation toolkit". In: *Nucl. Instruments Methods Phys. Res. Sect. A Accel. Spectrometers, Detect. Assoc. Equip.* (2003). ISSN: 01689002. DOI: 10.1016/S0168-9002(03)01368-8.

[167] M Cacciari, G. Salam, and G. Soyez. "The anti-k t jet clustering algorithm". In: *J. High Energy Phys.* (2008). ISSN: 11266708. DOI: 10.1088/1126-6708/2008/04/063. arXiv: 0802.1189.

[168] The CMS Collaboration. "Jet energy scale and resolution in the CMS experiment in pp collisions at 8 TeV". In: *J. Instrum.* (2017). ISSN: 17480221. DOI: 10.1088/1748-0221/12/02/P02014. arXiv: 1607.03663.

[169] M. Swartz. "CMS pixel simulations". In: *Nucl. Instruments Methods Phys. Res. Sect. A Accel. Spectrometers, Detect. Assoc. Equip.* 2003. DOI: 10.1016/S0168-9002(03)01757-1.

[170] V. Chiochia et al. "Simulation of heavily irradiated silicon pixel sensors and comparison with test beam measurements". In: *IEEE Trans. Nucl. Sci.* (2005). ISSN: 00189499. DOI: 10.1109/TNS.2005.852748.

[171] M. Swartz et al. "Observation, modeling, and temperature dependence of doubly peaked electric fields in irradiated silicon pixel sensors". In: *Nucl. Instruments Methods Phys. Res. Sect. A Accel. Spectrometers, Detect. Assoc. Equip.* (2006). ISSN: 01689002. DOI: 10.1016/j.nima.2006.05.002.

[172] D. Funke et al. "Parallel track reconstruction in CMS using the cellular automaton approach". In: *J. Phys. Conf. Ser.* 2014. DOI: 10.1088/1742-6596/513/5/052010.

[173] J. Schulte. "Tracking, alignment and b-tagging performance and prospects in CMS". In: *Proc. Sci.* 2018. DOI: 10.22323/1.321.0224.

[174] W. Adam et al. *Track Reconstruction in the CMS tracker*. Tech. rep. CMS-NOTE-2006-041. Geneva: CERN, Dec. 2006. URL: https://cds.cern.ch/record/934067.

[175] R. Fruhwirth. "Application of Kalman filtering to track and vertex fitting". In: *Nucl. Inst. Methods Phys. Res. A* (1987). ISSN: 01689002. DOI: 10.1016/0168-9002(87)90887-4.

[176] P. Billoir. "Progressive track recognition with a Kalman-like fitting procedure". In: *Comput. Phys. Commun.* (1989). ISSN: 00104655. DOI: 10.1016/0010-4655(89)90249-X.

[177] P. Billoir and S. Qian. "Simultaneous pattern recognition and track fitting by the Kalman filtering method". In: *Nucl. Inst. Methods Phys. Res. A* (1990). ISSN: 01689002. DOI: 10.1016/0168-9002(90)91835-Y.

[178] Rainer Mankel. "A concurrent track evolution algorithm for pattern recognition in the HERA-B main tracking system". In: *Nucl. Instruments Methods Phys. Res. Sect. A Accel. Spectrometers, Detect. Assoc. Equip.* (1997). ISSN: 01689002. DOI: 10.1016/S0168-9002(97)00705-5.

[179] S. Baffioni et al. "Electron reconstruction in CMS". In: *Eur. Phys. J. C* (2007). ISSN: 14346044. DOI: 10.1140/epjc/s10052-006-0175-5.

[180] *Particle-flow commissioning with muons and electrons from J/Psi and W events at 7 TeV*. Tech. rep. CMS-PAS-PFT-10-003. Geneva: CERN, 2010. URL: https://cds.cern.ch/record/1279347.

[181] W. Adam et al. "Reconstruction of electrons with the Gaussian-sum filter in the CMS tracker at the LHC". In: *J. Phys. G Nucl. Part. Phys.* (2005). ISSN: 09543899. DOI: 10.1088/0954-3899/31/9/N01. arXiv: 0306087 [physics].

[182] A. Hocker et al. "TMVA, the toolkit for multivariate data analysis with ROOT". In: *PHYSTAT LHC Work. Stat. Issues LHC Physics, PHYSTAT 2007 - Proc.* 2008. DOI: 10.22323/1.050.0040.

[183] P. Speckmayer et al. "The toolkit for multivariate data analysis, TMVA 4". In: *J. Phys. Conf. Ser.* 2010. DOI: 10.1088/1742-6596/219/3/032057.

[184] R. Brun and F. Rademakers. "ROOT - An object oriented data analysis framework". In: *Nucl. Instruments Methods Phys. Res. Sect. A Accel. Spectrometers, Detect. Assoc. Equip.* (1997). ISSN: 01689002. DOI: 10.1016/S0168-9002(97)00048-X.

[185] Y. Bengio, A. Courville, and P. Vincent. "Representation learning: A review and new perspectives". In: *IEEE Trans. Pattern Anal. Mach. Intell.* (2013). ISSN: 01628828. DOI: 10.1109/TPAMI.2013.50. arXiv: 1206.5538.

[186] P. Baldi, P. Sadowski, and D. Whiteson. "Searching for exotic particles in high-energy physics with deep learning". In: *Nat. Commun.* (2014). ISSN: 20411723. DOI: 10.1038/ncomms5308. arXiv: 1402.4735.

[187] C. Zhang et al. "Understanding deep learning requires rethinking generalization". In: *5th Int. Conf. Learn. Represent. ICLR 2017 - Conf. Track Proc.* 2017. arXiv: 1611.03530.

[188] B. Csaji. "Approximation with artificial neural networks". In: *MSc. thesis* (2001). DOI: 10.1.1.101.2647.

[189] J. Sola and J. Sevilla. "Importance of input data normalization for the application of neural networks to complex industrial problems". In: *IEEE Trans. Nucl. Sci.* (1997). ISSN: 00189499. DOI: 10.1109/23.589532.

[190] S. Ioffe and C. Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift". In: *32nd Int. Conf. Mach. Learn. ICML 2015*. 2015. ISBN: 9781510810587. arXiv: 1502.03167.

[191] S. Bengio. "Sharing Representations for Long Tail Computer Vision Problems". In: 2015. DOI: 10.1145/2818346.2818348.

[192] H. He and E. Garcia. "Learning from imbalanced data". In: *IEEE Trans. Knowl. Data Eng.* (2009). ISSN: 10414347. DOI: 10.1109/TKDE.2008.239.

[193] Jonathon Byrd and Zachary C. Lipton. "What is the effect of importance weighting in deep learning?" In: *36th Int. Conf. Mach. Learn. ICML 2019*. 2019. ISBN: 9781510886988. arXiv: 1812.03372.

[194] D. Soudry et al. "The implicit bias of gradient descent on separable data". In: *6th Int. Conf. Learn. Represent. ICLR 2018 - Conf. Track Proc.* 2018.

[195] S. Gunasekar et al. "Implicit bias of gradient descent on linear convolutional networks". In: *Adv. Neural Inf. Process. Syst.* 2018. arXiv: 1806.00468.

[196] T. O'Malley et al. *Keras Tuner*. https://github.com/keras-team/keras-tuner. 2019.

[197] L. Li et al. "Hyperband: A novel bandit-based approach to hyperparameter optimization". In: *J. Mach. Learn. Res.* (2018). ISSN: 15337928. arXiv: 1603.06560.

[198] J. Bergstra and Y. Bengio. "Random search for hyper-parameter optimization". In: *J. Mach. Learn. Res.* (2012). ISSN: 15324435.

[199] P. Ramachandran, B. Zoph, and Q. Le. "Searching for activation functions". In: *6th Int. Conf. Learn. Represent. ICLR 2018 - Work. Track Proc.* 2018. arXiv: 1710.05941.

[200] B. Ramachandran P.and Zoph and Q. Le. "Swish: a Self-Gated Activation Function". In: *arXiv:1710.05941v1* (2017). arXiv: 1710.05941.

[201] J. et al. Andersen. "Handbook of LHC Higgs Cross Sections: 3. Higgs Properties". In: (July 2013). Ed. by S Heinemeyer et al. DOI: 10.5170/CERN-2013-004. arXiv: 1307.1347 [hep-ph].

[202] CMS collaboration. *CMS Luminosity - Public Results*. https://twiki.cern.ch/twiki/bin/view/CMSPublic/LumiPublicResults. 2021.

[203] "Search for charged Higgs bosons in the $H^{\pm} \to \tau^{\pm}\nu$ decay channel in proton-proton collisions at $\sqrt{s} = 13$ TeV". In: *Journal of High Energy Physics* 2019.7 (July 2019). ISSN: 1029-8479. DOI: 10.1007/jhep07(2019)142. URL: http://dx.doi.org/10.1007/JHEP07(2019)142.

[204] J. Alwall et al. "The automated computation of tree-level and next-to-leading order differential cross sections, and their matching to parton shower simulations". In: *J. High Energy Phys.* (2014). ISSN: 10298479. DOI: 10.1007/JHEP07(2014)079. arXiv: 1405.0301.

[205] C. Degrande et al. "Accurate predictions for charged Higgs production: Closing the mH mt window". In: *Phys. Lett. Sect. B Nucl. Elem. Part. High-Energy Phys.* (2017). ISSN: 03702693. DOI: 10.1016/j.physletb.2017.06.037.

[206] P. Artoisenet et al. "Automatic spin-entangled decays of heavy resonances in Monte Carlo simulations". In: *J. High Energy Phys.* (2013). ISSN: 10298479. DOI: 10.1007/JHEP03(2013)015. arXiv: 1212.3460.

[207] S. Alioli et al. “A general framework for implementing NLO calculations in shower Monte Carlo programs: The POWHEG BOX”. In: *J. High Energy Phys.* (2010). ISSN: 10298479. DOI: 10.1007/JHEP06(2010)043. arXiv: 1002.2581.

[208] P. Nason. “A new method for combining NLO QCD with shower Monte Carlo algorithms”. In: *J. High Energy Phys.* (2004). ISSN: 10298479. DOI: 10.1088/1126-6708/2004/11/040. arXiv: 0409146 [hep-ph].

[209] S. Frixione, G. Ridolfi, and P. Nason. “A positive-weight next-to-leading-order Monte Carlo for heavy flavour hadroproduction”. In: *J. High Energy Phys.* (2007). ISSN: 11266708. DOI: 10.1088/1126-6708/2007/09/126. arXiv: 0707.3088.

[210] T. Jezo et al. “An NLO+PS generator for tt and Wt production and decay including non-resonant and interference effects”. In: *Eur. Phys. J. C* (2016). ISSN: 14346052. DOI: 10.1140/epjc/s10052-016-4538-2. arXiv: 1607.04538.

[211] R. Frederix and S. Frixione. “Merging meets matching in MC@NLO”. In: *J. High Energy Phys.* (2012). ISSN: 10298479. DOI: 10.1007/JHEP12(2012)061. arXiv: 1209.6215.

[212] S. Alioli et al. “NLO single-top production matched with shower in POWHEG: S- and t-channel contributions”. In: *J. High Energy Phys.* (2009). ISSN: 11266708. DOI: 10.1088/1126-6708/2009/09/111. arXiv: 0907.4076.

[213] E. Re. “Single-top Wt-channel production matched with parton showers using the POWHEG method”. In: *Eur. Phys. J. C* (2011). ISSN: 14346052. DOI: 10.1140/epjc/s10052-011-1547-z. arXiv: 1009.2450.

[214] J. Alwall et al. “Comparative study of various algorithms for the merging of parton showers and matrix elements in hadronic collisions”. In: *Eur. Phys. J. C* (2008). ISSN: 14346044. DOI: 10.1140/epjc/s10052-007-0490-5. arXiv: 0706.2569.

[215] The CMS Collaboration. “Event generator tunes obtained from underlying event and multiparton scattering measurements”. In: *Eur. Phys. J. C* (2016). ISSN: 14346052. DOI: 10.1140/epjc/s10052-016-3988-x. arXiv: 1512.00815.

[216] *Investigations of the impact of the parton shower tuning in Pythia 8 in the modelling of* $\mathrm{t\bar{t}}$ *at* $\sqrt{s} = 8$ *and 13 TeV*. Tech. rep. Geneva: CERN, 2016. URL: https://cds.cern.ch/record/2235192.

[217] J. Allison et al. “Geant4 developments and applications”. In: *IEEE Trans. Nucl. Sci.* (2006). ISSN: 00189499. DOI: 10.1109/TNS.2006.869826.

[218] *Procedure for the LHC Higgs boson search combination in Summer 2011*. Tech. rep. Geneva: CERN, Aug. 2011. URL: https://cds.cern.ch/record/1379837.

[219] The CMS Collaboration. “Performance of reconstruction and identification of τ leptons decaying to hadrons and $\nu\tau$ in pp collisions at s=13 TeV”. In: *J. Instrum.* (2018). ISSN: 17480221. DOI: 10.1088/1748-0221/13/10/P10005.

[220] *Jet Performance in pp Collisions at 7 TeV*. Tech. rep. CMS-PAS-JME-10-003. Geneva: CERN, 2010. URL: http://cds.cern.ch/record/1279362.

[221] The CMS Collaboration. “Identification of b-quark jets with the CMS experiment”. In: *J. Instrum.* (2013). ISSN: 17480221. DOI: 10.1088/1748-0221/8/04/P04013. arXiv: 1211.4462.

[222] P. Baldi et al. “Parameterized neural networks for high-energy physics”. In: *Eur. Phys. J. C* (2016). ISSN: 14346052. DOI: 10.1140/epjc/s10052-016-4099-4.

[223] G. Kasieczka and D. Shih. "Robust jet classifiers through distance correlation". In: *Phys. Rev. Lett.* (2020). ISSN: 10797114. DOI: 10.1103/PhysRevLett.125.122001.

[224] P. Windischhofer, M. Zgubic, and D. Bortoletto. *Preserving physically important variables in optimal event selections: A case study in Higgs physics*. 2019. arXiv: 1907.02098.

[225] I. Moult, B. Nachman, and D. Neill. "Convolved substructure: analytically decorrelating jet substructure observables". In: *J. High Energy Phys.* (2018). ISSN: 10298479. DOI: 10.1007/JHEP05(2018)002. arXiv: 1710.06859.

[226] L. Bradshaw et al. *Mass Agnostic Jet Taggers*. 2019. arXiv: 1908.08959.

[227] E. Pekkarinen. "Data-driven measurement of the background with misidentified tau leptons in a search for charged Higgs bosons; Väärintunnistettuja tau leptoneita sisältävän taustan datalähtöinen mittaus varattujen Higgsin bosonien etsinnässä". en. G2 Pro gradu, diplomityö. 2015-03-31, pp. 81+9. URL: http://urn.fi/URN:NBN:fi:aalto-201504082237.

[228] The CMS Collaboration. "Performance of CMS muon reconstruction in pp collision events at s = 7TeV". In: *J. Instrum.* (2012). ISSN: 17480221. DOI: 10.1088/1748-0221/7/10/P10002. arXiv: 1206.4071.

[229] The CMS Collaboration. "Identification of heavy-flavour jets with the CMS detector in pp collisions at 13 TeV". In: *J. Instrum.* (2018). ISSN: 17480221. DOI: 10.1088/1748-0221/13/05/P05011. arXiv: 1712.07158.

[230] *Performance of missing transverse momentum in pp collisions at sqrt(s)=13 TeV using the CMS detector*. Tech. rep. CMS-PAS-JME-17-001. Geneva: CERN, 2018. URL: https://cds.cern.ch/record/2628600.

[231] J. Butterworth et al. "PDF4LHC recommendations for LHC Run II". In: *J. Phys. G Nucl. Part. Phys.* (2016). ISSN: 13616471. DOI: 10.1088/0954-3899/43/2/023001. arXiv: 1510.03865.

[232] *CMS Luminosity Measurements for the 2016 Data Taking Period*. Tech. rep. CMS-PAS-LUM-17-001. Geneva: CERN, 2017. URL: https://cds.cern.ch/record/2257069.

[233] M. Stoye. "Deep learning in jet reconstruction at CMS". In: *J. Phys. Conf. Ser.* 2018. DOI: 10.1088/1742-6596/1085/4/042029.

[234] ATLAS collaboration. "Search for charged Higgs bosons decaying via $H^{\pm} \to \tau \pm \nu_{\tau}$ in the τ+jets and τ+lepton final states with 36 fb-1 of pp collision data recorded at $\sqrt{s} = 13$ TeV with the ATLAS experiment". In: *Journal of High Energy Physics* 2018.9 (Sept. 2018). ISSN: 1029-8479. DOI: 10.1007/jhep09(2018)139. URL: http://dx.doi.org/10.1007/JHEP09(2018)139.

[235] N. Smith et al. "Coffea Columnar Object Framework For Effective Analysis". In: *EPJ Web Conf.* (2020). DOI: 10.1051/epjconf/202024506012.